

普通高等教育“十一五”国家级规划教材

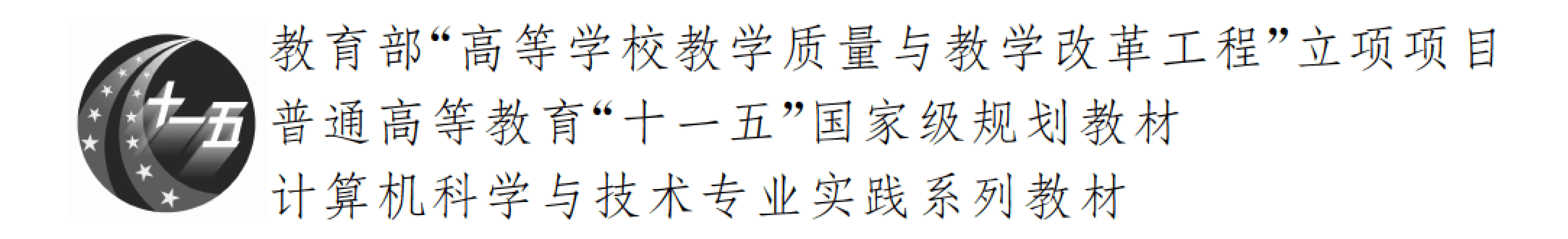
教育部“高等学校教学质量与教学改革工程”立项项目

张兴会 等编著

数据仓库与数据挖掘技术

计算机科学与技术专业实践系列教材

清华大学出版社



教育部"高等学校教学质量与教学改革工程"立项项目
普通高等教育"十一五"国家级规划教材
计算机科学与技术专业实践系列教材

数据仓库与数据挖掘技术

张兴会　等编著

清华大学出版社
北　京

内 容 简 介

数据仓库与数据挖掘是计算机专业和其他一些与计算机技术关系密切专业必修的核心课程。本书系统地介绍了数据仓库和数据挖掘的基本概念、相关知识和基本方法，每种数据挖掘方法都有详尽的实例描述和具体实现步骤。

本书结构严谨，条理清晰，语言浅显易懂，循序渐进地表达了知识内容；本书坚持理论与实际相结合，概念和具体方法相结合，使知识具体化，生动化；实例实现的过程建立在 SQL 2005 数据挖掘软件的基础上，以帮助读者在学习后达到学以致用的目的。

本书可以作为计算机类、信息类等相关专业本科生数据挖掘课程的教材，也可以作为其他专业技术人员的自学参考书。

图书在版编目（CIP）数据

数据仓库与数据挖掘技术/张兴会等编著．—北京：清华大学出版社，2011.6
（计算机科学与技术专业实践系列教材）
ISBN 978-7-302-24701-2

Ⅰ．①数…　Ⅱ．①张…　Ⅲ．①数据库系统—高等学校—教材 ②数据采集—高等学校—教材　Ⅳ．①TP311.13 ②TP274

中国版本图书馆 CIP 数据核字(2011)第 018775 号

责任编辑：汪汉友
责任校对：时翠兰
责任印制：何　芊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　　**印　　张**：14.25　　**字　　数**：344 千字
版　　次：2011 年 6 月第 1 版　　**印　　次**：2011 年 6 月第 1 次印刷
印　　数：1～4000
定　　价：24.00 元

产品编号：034551-01

普通高等教育"十一五"国家级规划教材
计算机科学与技术专业实践系列教材

前　　言

随着信息技术的高速发展,数据量的积累急剧增长,数据挖掘是为顺应这种需要而发展起来的数据处理技术,是知识发现(Knowledge Discovery in Database)的关键步骤。数据挖掘涉及比较多的数学基础知识,如何深入浅出地将这些知识及其应用方法介绍给学生是编写数据挖掘教材的关键所在。

为此,本书在编写时力求突出以下特征:

(1) 采用尽可能浅显易懂的语言表达知识内容;

(2) 理论与实际相结合,概念和方法相结合,使知识具体化,实用化;

(3) 实例是通过数据挖掘软件 SQL Server 2005 完成的;

(4) 每章最后结合实例,理论联系实际,帮助学生达到学以致用的效果。

本书共 11 章,包括 4 个主要部分,具体内容如下。

第一部分:第 1 章为数据挖掘和数据仓库概述,简要介绍了数据挖掘和数据仓库的发展趋势、基本概念等相关知识。

第二部分:第 2 章和第 3 章详细介绍了数据仓库的基本概念、相关知识,以及联机分析处理技术的基本方法和实例的具体实现。

第三部分:第 4 章～第 10 章详细介绍了关联规则方法、决策树方法、统计学习方法、神经网络方法、聚类分析、粗糙集方法等方法的相关知识和实例的具体实现。

第四部分:第 11 章介绍了一些复杂结构的数据挖掘以及数据挖掘的发展。

本书的亮点为,每章的最后一节都是本章理论方法的一个具体实现,便于读者深入掌握。读者可以根据自己的需要选择学习相关内容。本书可以作为计算机类、信息类等相关专业本科生数据挖掘课程的教材,也可以作为其他专业技术人员的自学参考书。

信息处理技术是信息科学、应用数学发展的一个重要分支,在教学中,主要通过理论教学、实验教学、课程设计等教学环节来提高学生的实践技能和应用水平,这样的教学方法也是天津职业技术师范大学长期为社会培养高素质职教师资和应用型高级专门人才过程中总结出来的一种行之有效的教学方法。为了使理论和实际相结合,使基本概念和知识与具体的方法、工具相结合,达到学以致用的效果,体现应用型大学手脑并用的办学理念,作者还特别编写了一本与本教材相配合使用的《数据仓库与数据挖掘工程实例》辅助教材,该教材共包括 10 个工程案例,介绍了利用数据挖掘与数据仓库工具如何建立数据仓库、如何进行数据预处理和进行数据挖掘等,目的就是希望通过通俗易懂的语言和详细的工程实例分析,使学生能够较好地掌握数据挖掘与数据仓库的理论知识和构建模型的操作过程,进一步提高学生对信息进行管理和利用的能力。

本书由张兴会统稿,王明春、郑晓艳、刘玲、刘新钰、童勇木参加了本书的编写、图表绘

制、模型构建、软件调试等工作。在本书编写过程中，安淑芝教授提出了宝贵的修改意见。另外，本书还参阅和引用了许多专家和学者的文献资料，在此表示衷心的感谢。

由于作者水平和能力有限，新技术的发展和更新较快，书中的不妥之处，欢迎读者批评指正。作者邮箱：xhzhang@tute.edu.cn。

作者

2011 年 4 月于天津

目　　录

第1章　数据挖掘和数据仓库概述

随着计算机技术和网络技术的发展,数据量急剧增长。人类处于信息爆炸的时代,被淹没在数据海洋之中。如何有效地组织和存储数据,如何从数据海洋中及时发现有用的知识、提高信息利用率,成为人们亟待解决的问题。但是,仅以目前数据库系统的录入、查询、统计等功能,无法发现数据中存在的关系和规则,无法根据现有的数据预测未来的发展趋势。正是在这样的背景下,数据挖掘(data mining,DM)技术应运而生,并越来越显示出强大的生命力。

数据挖掘技术的发展催生决策分析数据环境的改变,而传统的数据库管理系统因自身的局限性无法满足决策支持系统的要求,具体表现为:不能满足数据成几何级数增长的需要,不同部分的数据难以集成,访问数据的响应性能不断降低。要想使数据能够发挥其最佳效用,更好地为用户服务,数据必须经过严格的准备、组织和显示等步骤。因此,一种适用于决策支持系统的数据组织与管理技术——数据仓库(data warehouse,DW)技术应运而生,并逐渐成为支持分析与决策的重要技术。

1.1　数据挖掘引论

1.1.1　数据挖掘的由来

数据挖掘经历了逐渐演变的过程。在电子化数据处理的初期,人们就试图通过某些方法实现自动决策支持,于是机器学习成为关注的焦点。机器学习的过程就是将一些已知的并已被成功解决的问题作为范例输入计算机,机器通过学习这些范例,总结并生成相应的规则,这些规则具有通用性,使用它们可以解决某一类问题。机器学习的研究最早始于20世纪60年代,比较典型的结果有Rosenblate的感知机、Sammel的西洋跳棋程序。

随着神经网络等技术的形成和发展,人们的注意力逐渐转向知识工程。知识工程不同于机器学习,不是为计算机输入范例,由其生成出规则,而是直接为计算机输入已被代码化的规则,计算机通过使用这些规则来解决某些问题,如专家系统就是这种方法所得到的成果。

20世纪80年代,在新的神经网络等理论的指导下,重新回到机器学习的方法上,并将其成果应用于处理大型商业数据库,如Michelski等人的AQ11系统(1980年)、Quiulan的ID3(1983年)决策树方法、Rumelhart等人研制的反向传播神经网络BP模型(1985年)、Langley等人的BACON系统(1987年)等,这些显著成果的出现,使机器学习逐渐成为人工智能的主要学科方向之一。

1989年,在美国底特律召开的第十一届国际联合人工智能学术会议上首次提到知识发现(knowledge discovery in database,KDD)这一概念;1993年,美国电气电子工程师学会(IEEE)的知识与数据工程(knowledge and data engineering)会刊出版TKDD技术专刊,发

表的论文和摘要体现了当时知识发现的最新研究成果和动态。

随着来自各个领域的研究和应用开发不断增多，1995 年，在加拿大蒙特利尔召开了首届 KDD 国际学术年会，数据挖掘技术被分为工程领域的数据挖掘与科研领域的知识发现。由于把数据库中的“数据”形象地比喻为矿床，“数据挖掘”一词很快流传开来。此后，此类会议每年召开一次，数量和规模逐渐扩大，从专题研讨会一直发展到国际学术大会，并成为当前计算机领域的研究热点。目前，对 KDD 的研究主要围绕理论、技术和应用这三个方面展开。

1.1.2 数据挖掘的定义

数据挖掘就是从大量的、不完全的、有噪声的、模糊的、随机的数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。数据挖掘应该更正确地命名为“从数据中挖掘知识”。还有很多和这一术语相似的术语，如知识发现、数据分析、数据融合以及决策支持等。人工智能领域习惯称之为知识发现，而数据库领域习惯称之为数据挖掘。

用于数据挖掘的原始数据可以是结构化的，如关系数据库中的数据；也可以是半结构化的，如文本、图形、图像数据等。数据挖掘的方法可以是数学的，也可以是非数学的；可以是演绎的，也可以是归纳的。挖掘出的知识可以被用于信息管理、查询优化、决策支持、过程控制等；还可以用于数据自身的维护。

数据挖掘是一个完整的过程，其一般步骤如图 1-1 所示。

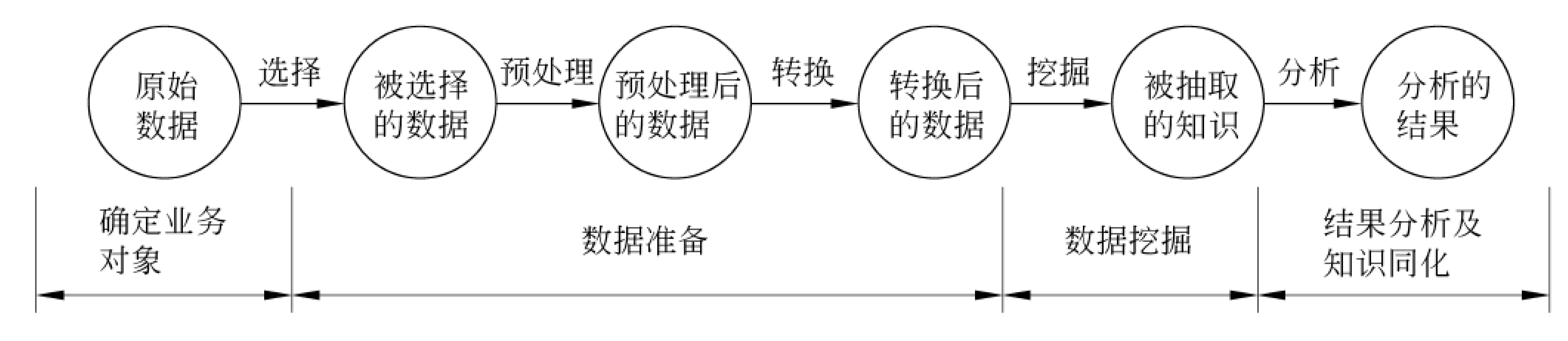


图 1-1 数据挖掘的过程

1. 确定业务对象

在开始数据挖掘之前，最基础的就是理解数据和实际的业务，在这个基础上提出问题，对目标有明确的定义。认清数据挖掘的目的是数据挖掘的重要一步，因此必须清晰地定义出业务对象。

2. 数据准备

数据准备是保证数据挖掘得以成功的先决条件，数据准备在整个数据挖掘过程中占有很大的工作量，大约是整个数据挖掘工作量的 60%。数据准备包括数据选择、数据预处理和数据转换。

(1) 数据选择。就是搜索所有与业务对象有关的内部和外部数据信息，获取原始的数据，从中选择出适用于数据挖掘应用的数据，建立数据挖掘库。

(2) 数据预处理。由于数据可能是不完全的、有噪声的、随机的、复杂的，数据预处理就要对数据进行初步的整理，清洗不完全的数据，为进一步的分析作准备。

(3) 数据转换。数据转换是构建面向分析的数据存储模式的关键，在转换过程中数据

会被格式化，并加载到适合分析的存储环境中。常见的数据转换问题包括：数据类型转换、对象名转换、数据编码转换、表结构转换。

3. 数据挖掘

数据挖掘就是对所得到的经过转换的数据进行挖掘，除了选择合适的挖掘算法外，其余工作应能自动地完成。

4. 结果分析与知识同化

结果分析就是对挖掘结果进行解释并评估，其使用的分析方法一般应根据数据挖掘操作而定，目前通常应用可视化技术。知识同化就是将分析所得到的知识集成到业务信息系统的组织结构中去。

1.1.3 数据挖掘的功能

数据挖掘具体功能主要有以下几个方面。

1. 概念描述

概念描述(concept description)，就是对某类对象的内涵进行描述，并概括这类对象的有关特征。具体的描述分为特征性(characterization)描述和区别性(discrimination)描述。前者用于描述某类对象的共同特征，后者用于描述不同类对象之间的区别。

描述数据允许数据在多个抽象层概化，便于用户考察数据的一般行为。

2. 关联分析

数据关联是数据中存在的一类重要的可被发现的知识，若两个或多个变量间存在着某种规律性，就称为关联。关联可分为简单关联、时序关联、因果关联。关联分析(association analysis)是从大量的数据中发现项集之间有趣的联系、相关关系或因果结构，以及项集的频繁模式。

3. 分类与预测

(1) 分类(classification)。分类是数据挖掘中的一项非常重要的任务。分类的目的是提出一个分类函数或者分类模型，该模型能把数据库中的数据项映射到给定类别中的一个。构造分类器，需要有一个训练样本数据集作为输入。

(2) 预测(prediction)。预测是利用历史数据建立模型，再运用最新数据作为输入值，获得未来变化的趋势或者评估给定样本可能具有的属性值或值的范围。

4. 聚类分析

(1) 聚类(clustering)。聚类是根据数据的不同特征，将其划分为不同的数据类。其目的是使得属于同一类别的个体之间的距离尽可能小，而不同类别的个体间的距离尽可能大。

(2) 聚类与分类的区别如下：分类需要预先定义类别和训练样本；而聚类分析直接面向源数据，没有预先定义好的类别和训练样本，所有记录都根据彼此相似程度加以归类。

5. 偏差分析

偏差分析(deviation analysis)又称为比较分析，是对差异和极端特例的描述，揭示事物偏离常规的异常现象，其基本思想是寻找观测结果与参照值之间有意义的差别。偏差包括分类中的反常实例、不满足规则的特例、观测结果对模型预测的偏差、量值随时间的变化等。

1.1.4 数据挖掘的常用方法

1. 聚类分析

聚类分析(clustering analysis)是一个比较活跃的数据挖掘研究领域,源于统计学、生物学以及机器学习等。聚类生成的组叫簇,簇是数据对象的集合。聚类分析的过程就是使同一个簇内的任意两个对象之间具有较高的相似性,不同簇的两个对象之间具有较高的相异性。

用于数据挖掘的聚类分析有划分的方法、层次的方法、基于密度的方法、基于网格的方法和基于模型的方法等。

2. 决策树

决策树(decision tree)主要应用于分类和预测,提供了一种展示类似在什么条件下会得到什么值这类规则的方法。决策树分为分类树和回归树两种,分类树对离散变量做决策,回归树对连续变量做决策。

决策树是一个类似于流程图的树结构,树的最顶层结点是根结点,中间的结点是内部结点,末梢的结点是叶结点,其中根结点是整个数据集合空间,每个内部结点表示在一个属性上的测试,每个分支代表一个测试输出,每个叶结点代表类或类分布。

建立决策树的过程,即树的生长过程是不断地把数据进行切分的过程,每次切分对应一个问题,也对应着一个结点。对每个切分都要求分成的组之间的"差异"最大。各种决策树算法之间的主要区别是"差异"衡量方式的区别。数据挖掘中决策树是一种经常用到的技术,常用的算法有 CHAID、CART、Quest、ID3 和 C4.5 等。

3. 人工神经网络

人工神经网络(artificial neural network,ANN)是一类比较新的计算模型,它是模仿人脑神经网络的结构和某些工作机制而建立的一种计算模型。这种计算模型的特点是利用大量的简单计算单元(即神经元)连成网络,来实现大规模并行计算。神经网络的工作机理是通过学习,来改变神经元之间的连接强度。由于人工神经网络具有自我组织和自我学习等特点,能解决许多其他方法难以解决的问题,因此得到较普遍的应用。

人工神经网络主要有前馈式网络、反馈式网络和自组织网络。

4. 粗糙集

粗糙集(rough set)是一种处理不确定、不完备数据和不精确问题的新的数学理论。粗糙集理论建立在分类机制的基础上,将知识理解为对数据的划分,并引入上近似(upper approximation)和下近似(lower approximation)等概念来刻画知识的不确定性和模糊性。模糊集和概率统计方法是处理不确定信息的常用方法,但这些方法需要一些数据的附加信息或先验知识,如模糊隶属函数和概率分布等,这些信息有时并不容易得到。粗糙集分析方法仅利用数据本身提供的信息,无须任何先验知识。

5. 关联规则挖掘

关联规则挖掘(association rule mining)是数据挖掘中最活跃的研究方法之一,最早由 Agrawal 等人提出(1993 年)。最初的动机是针对购物篮分析问题提出的,其目的是发现交易数据库中不同商品(项)之间的联系,由这些规则找出顾客购买行为模式,如购买了某一商品对购买其他商品的影响。发现这样的规则可以应用于商品货架设计、库存安排以及根据

购买模式对用户进行分类。

关联规则的基本思想：一是找到所有支持度大于最小支持度的频繁项集，即频集；二是使用第一步找到的频集产生期望的规则。其核心方法是基于频集理论的递推方法。关联规则挖掘的主要算法包含关联发现、序列模式发现、时序发现等。

6. 统计分析

统计分析(statistics analysis)是从事物的外在数量上的表现去推断该事物可能的规律。科学的规律性一般总是隐藏得比较深，最初总是从其数量表现上通过统计分析看出一些线索，然后提出一定的假说或学说，做进一步深入的理论研究。当理论研究提出一定的结论时，往往还需要在实践中加以验证，即观测一些自然现象或专门安排的实验所得资料是否与理论相符，在多大程度上相符，偏离可能是朝哪个方向，等等。

常见的统计分析有回归分析(多元回归、自回归)、判别分析(贝叶斯判别、费歇尔判别、非参数判别)以及探索性分析(主元分析、相关分析)等。

1.2 数据仓库引论

1.2.1 数据仓库的产生与发展

随着市场竞争的日趋激烈，信息对企业的生存、发展、壮大起着越来越重要的作用。由于计算机技术的普遍应用，承载信息的数据随着时间的推移而不断增长，并且分布在不同的系统平台上，具有多种存储形式。能否从纷繁复杂、大量沉淀的数据环境中得到有用的决策信息，已成为企业生存、发展、壮大的重要环节。

基于上述的需求，在 20 世纪 80 年代出现了数据仓库的思想。1988 年，为解决全企业集成问题，IBM 爱尔兰公司的 Barry Devlin 和 Paul Murphy 第一次提出了“信息仓库”的概念，其定义为：“一个结构化的环境，能支持最终用户管理其全部的业务，并支持信息技术部门保证数据质量”。在 20 世纪 90 年代初期，数据仓库的基本原理、框架架构，以及分析系统的主要原则都已经确定，主要技术包括关系型数据存取、网络、C/S 架构和图形化界面。一些前沿的公司已经开始建立数据仓库。

1992 年，美国著名的信息工程学家 William H. Inmon 在《建立数据仓库》(Building the Data Warehouse)一书中首先系统地阐述了关于数据仓库的思想、理论。该书不仅说明为什么要建数据仓库、数据仓库能带来什么，更重要的是，Inmon 第一次提供了如何建设数据仓库的指导性意见。该书定义了数据仓库非常具体的原则，即：数据仓库是面向主题的、集成的、包含历史的、不可更新的、面向决策支持的、面向全企业的、最明细的数据存储、数据快照式的数据获取等。这些原则到现在仍然是指导数据仓库建设的最基本原则，因此，William H. Inmon 被人们尊称为“数据仓库之父”。

数据仓库的盛行始于 1995 年，而且其作为数据库的高端扩展技术一直是一大热点。IBM 所推崇的商业智能(BI)，其核心就是数据仓库；微软的 SQL Server 7.0 已经绑定了 OLAP 服务器，将数据仓库功能集成到数据库中，并建立了数据仓库联盟；Oracle 公司也有自己的 Oracle Express 系列 OLAP 产品用来提供决策支持。

从目前形势看，数据仓库已成为继因特网之后，信息社会中获得企业竞争优势的关键。

据美国 Meta Group 市场调查机构的资料表明,《幸福》杂志所列的全球 2000 家大公司中,已有 99%将因特网和数据仓库这两项技术都列入企业计划。

1.2.2 数据仓库的定义

William H. Inmon 在《Building the Data Warehouse》一书中指出,“数据仓库是面向主题的、集成的、具有时间特征的、稳定的数据集合,用以支持经营管理中的决策制定过程”。由于 William H. Inmon 本人在数据仓库发展中的重要作用,他对数据仓库的上述描述不断被其他文献引用,相对成了一种权威的定义。

与传统数据库相比,数据仓库虽然是从数据库发展而来的,但是两者在许多方面都存在着相当大的差异。从数据存储内容看,数据库只存放当前值,而数据仓库则存放历史值;数据库中数据的目标是面向业务操作人员,提供事务处理的支持,而数据仓库则是面向中高层管理人员,提供决策支持;数据库内的数据是动态变化的,只要有业务发生,数据就会被更新,而数据仓库则是静态的历史数据,只能定期添加;数据库中的数据结构比较复杂,用各种数据结构来满足业务处理系统的需要,而数据仓库中的数据结构则较为简单;数据库中数据的访问频率高,但是访问数据的量少,而数据仓库的访问频率低,但是数据访问量要远高于数据库;数据库在访问数据时要求响应速度很快,其响应时间一般要求在数秒以内,而数据仓库的响应时间可能长达数小时。

1.2.3 数据仓库与数据挖掘的联系与区别

1. 数据仓库与数据挖掘的联系

数据挖掘和数据仓库作为决策支持新技术,在近十年来得到迅速发展。作为数据挖掘对象,数据仓库技术的产生和发展为数据挖掘技术开辟了新的战场,同时也提出了新的要求和挑战。数据仓库和数据挖掘是相互影响、相互促进的。二者的联系可以概括为以下几点。

(1) 数据仓库为数据挖掘提供了更好的、更广泛的数据源。在数据仓库中,集成和存储着来自异质信息源的数据,而这些信息源本身就可能是一个规模庞大的数据库。同时数据仓库存储了大量长时间的历史数据(5~10 年),这使得人们可以进行数据长期趋势的分析,这为决策者的长期决策行为提供了支持。数据仓库中数据在时间轴上的纵深性是数据挖掘不能回避的一个新难点。

(2) 数据仓库为数据挖掘提供了新的支持平台。数据仓库的发展不仅为数据挖掘开辟了新的空间,也对数据挖掘技术提出了更高的要求。数据仓库一般设计成只读方式,数据仓库的更新由专门的机制保证。数据仓库对查询的强大支持使数据挖掘效率更高,挖掘过程可以做到实时交互,使决策者的思维保持连续,有可能挖掘出更深入、更有价值的知识。

(3) 数据仓库为更好地使用数据挖掘工具提供了方便。数据仓库的建立,充分考虑数据挖掘的要求。用户可以通过数据仓库服务器得到所需的数据,形成中间数据库,利用数据挖掘方法进行挖掘,获得知识。数据挖掘要面对的是关系更复杂的企业全局模式的知识,数据仓库为数据挖掘集成了企业内各部门的全面的、综合的数据。而且,数据仓库机制大大降低了数据挖掘的障碍,一般进行数据挖掘要花大量的精力在数据准备阶段。数据仓库中的

数据已经被充分收集起来，进行了整理、合并，有些还进行了初步的分析处理。这样，数据挖掘的注意力能够更集中于核心处理阶段。另外，数据仓库中对数据不同粒度的集成和综合，更有效地支持了多层次、多种知识的挖掘。

(4) 数据挖掘为数据仓库提供了更好的决策支持。企业领导的决策分析要求系统能够提供更高层次的决策辅助信息，从这一点上讲，基于数据仓库的数据挖掘能更好地满足战略决策的要求。数据挖掘对数据仓库中的数据进行模式抽取和发现知识，这些正是数据仓库所不能提供的。

(5) 数据挖掘对数据仓库的数据组织提出了更高的要求。数据仓库作为数据挖掘的对象，要为数据挖掘提供更多、更好的数据。其数据的设计、组织都要考虑到数据挖掘的一些要求。

(6) 数据挖掘还为数据仓库提供了广泛的技术支持。数据挖掘的可视化技术、统计分析技术等都为数据仓库提供了强有力的技术支持。

总之，数据仓库在纵向和横向上都为数据挖掘提供了更广阔的活动空间。数据仓库完成数据的收集、集成、存储、管理等工作，数据挖掘面对的是经过初步加工的数据，使得数据挖掘能更专注于知识的发现；数据仓库所具有的面向主题、集成、时间特征、数据稳定等特点，对数据挖掘技术提出了更高的要求，而数据挖掘为数据仓库提供了更好的决策支持，促进了数据仓库技术的发展。可以说，数据挖掘和数据仓库技术要充分发挥潜力，就必须结合起来。

2. 数据仓库与数据挖掘的区别

数据仓库是一种存储技术，它包含大量的历史数据、当前的详细数据以及综合数据，它能为不同用户的不同决策需要提供所需的数据和信息。

数据挖掘是从人工智能机器学习中发展起来的，它研究各种方法和技术，从大量的数据中挖掘出有用的信息和知识。

1.3 数据挖掘的应用

随着人们对数据挖掘认识的深入，数据挖掘技术应用越来越广泛，成功的案例很多。某些具有特定的应用问题和应用背景的领域，最能体现数据挖掘的作用。目前，数据挖掘应用在金融业和保险业较多，也扩展到了其他应用领域，如零售业、医疗保健、运输业、行政司法等社会部门以及科学和工程研究单位。

1.3.1 数据挖掘的应用领域

1. 金融业

金融业可以用数据挖掘分析市场的动向、预测公司的营运能力和股价趋势等。

(1) 评估账户信用等级。金融业风险与效益并存，分析账户的信用等级对于降低风险、增加收益是非常重要的。利用数据挖掘技术进行信用评估，可以从已有的数据中分析得到信用评估的规则或标准，即得到“满足什么样条件的账户属于哪一类信用等级”，并将得到的规则或评估标准应用到对新账户的信用评估。

(2) 分析信用卡使用模式。通过数据挖掘技术分析信用卡的使用模式，可以知道：“什

么样的人使用信用卡属于什么样的模式”。一般在相当长的一段时间内，个人使用信用卡的习惯往往是较为固定的。因此，通过判别信用卡的使用模式，可以监测到信用卡的恶性透支行为，还可以根据信用卡的使用模式，识别“合法”用户。

（3）分析股票趋势。可以利用数据挖掘技术从股票交易的历史数据中得到股票交易的规则或规律。

（4）探测金融政策与金融行情关系。利用数据挖掘技术，可以从大量的历史记录中发现或挖掘出金融政策与金融业行情之间相互影响的更深层次的关联关系。

数据挖掘技术在金融业还可以对庞大的数据进行主成分分析，剔除无关的甚至是错误的、相互矛盾的数据“杂质”，以更有效地进行金融市场分析和预测，发现隐藏在数据后面不同的财政金融指数之间的联系。

2. 保险业

（1）确定保险金。对受险人员的分类有助于确定适当的保险金额度。通过数据挖掘可以有助于确定对不同行业、不同年龄段、不同社会层次人员保险金的额度。

（2）险种关联分析。使用数据挖掘技术，通过险种关联分析，可以预测购买了某种保险的人是否会同时购买另一种保险。

（3）其他预测。通过使用数据挖掘技术可以预测哪些行业、哪个年龄段、哪种社会层次的人会买哪种保险，或者预测哪类人容易买新的险种等。

3. 科学研究

（1）自然科学。数据挖掘技术对于高科技研究来说是必不可少的，主要功能是对大批量数据的处理。高科技研究的特点就是探索人类未知的秘密，而这正是数据挖掘的特长所在。不借助于数据挖掘技术，要从大量的、漫无头绪而且真伪难辨的数据和资料中提炼出对人类有用的信息，是非常困难的。

（2）社会科学。数据挖掘在社会科学研究领域的应用前景也越来越被人们所认识。社会科学的特点是从历史看未来，如从社会发展的历史进程中得出社会发展的规律，预测社会发展的趋势；或从人类发展的进程和人类社会行为的变化中寻求对人类行为规律的答案，从而求解各种各样的社会问题。

（3）信息科学。

① 电子商务。通过分析网站的参观者和购物者的购买浏览行为，可以给网站经营者提供很好的决策依据。例如，找出一条浏览频率高的路径之后，可以进一步从中分析用户走此路径的目的是为了查看哪种产品的相关信息，进而可以考虑在相关的网页中加强该产品的广告宣传，以刺激用户的购物欲望，增加销售量。

② 个性化服务。用户在网上浏览过程中总是会出现一些自己毫不关心的话题，解决方法就是要把以网站为中心转换为以用户为中心，提供个性化服务。个性化服务就是根据发现的用户喜好，动态地为用户定制所需的内容或提供浏览建议。实施的基本思路是，在 Web 挖掘的基础上，根据浏览页面内容之间的相关性，为用户提供个性化服务。尽可能使每个用户在浏览商业网站时，都有一种自己是该网站唯一用户的感觉；尽可能地迎合每个用户的浏览兴趣，并且不断调整网站内容来适应用户浏览兴趣的变化。例如，依据网站中的网页内容，将适合的网页推荐给适合的用户；根据客户的喜好程度来推荐物品。

③ 网络教学。网络教学与传统教学最大的差别在于教师无法直接和学生面对面接触，因而，在网络教学成效评估上，可利用数据挖掘技术挖掘网络教学中使用者的浏览信息，找出学生在学习过程中最常访问的网页后，分析学生的学习状况，进一步提升整个网络教学的品质。

4. 市场营销

企业以市场营销学的市场细分原理为基础，通过收集、加工和处理涉及消费者消费行为的大量信息，确定特定消费群体或个体的兴趣、消费习惯、消费倾向和消费需求，进而推断出相应消费群体或个体下一步的消费行为，然后以此为基础，对所识别出来的消费群体进行特定内容的定向营销，这与传统的不区分消费者对象特征的大规模营销手段相比，大大节省了营销成本，提高了营销效果，从而为企业带来更多的利润。

5. 客户关系管理

客户关系管理是数据挖掘在商务领域应用中的一个重要方面。数据挖掘已成为客户关系管理系统的必备功能和主要实现手段。

客户关系管理是指企业通过富有意义的交流沟通，理解并影响客户行为，最终实现提高客户获得、客户保留、客户忠诚和客户创利的目的。

客户关系管理数据分析包括以下 4 点内容。

(1) 整合存放在不同数据库中相互关联的原始数据，进行关联性查询。

(2) 对历史数据进行分析。从历史数据中选择不同的角度考察消费行为；评估客户价值，细分客户群；利用数据验证行业经验；针对不同的客户群发掘消费特点；定期地将原始数据抓取到与运营系统分离的数据仓库中并完成分析图表，确保有效地降低等待时间；平衡分析的灵活自定义和分析结果的反馈速度。

(3) 收益/客户消费预测。建立数据模型，对不同的客户群预测消费量；调整重要参数，估计对收益或利润的影响；对市场活动的效果进行预测；从不同的维度进行知识发现。

(4) 优化方法。利用数据模型进行优化，以确立适合的价格策略；通过设置商业规则，进行复杂的市场划分；平衡市场活动的费用和效益。

6. 其他领域

(1) 医疗。数据挖掘可用于病例、病人行为特征分析、药方管理等，以安排治疗方案、判断药方的有效性等。

(2) 司法。数据挖掘可用于案件调查、案例分析、犯罪监控等，还可用于犯罪行为特征的分析。

(3) 工业部门。数据挖掘技术在工业部门应用于故障诊断、生产过程优化等。如制造业在质量控制、制造过程中，找出影响产品品质的最大因素及提高作业流程的效率等方面，都可以应用数据挖掘技术。

1.3.2 数据挖掘案例

1. 竞技运动中的数据挖掘

美国著名的 NBA 篮球队的教练，利用 IBM 公司提供的数据挖掘工具临场决定替换队员。若读者是 NBA 的教练，那么靠什么来带领球队取得胜利呢？当然，最容易想到的是全场紧逼、交叉扯动和快速抢断等具体的战术和技术。今天，NBA 的教练又有了新式武器：

数据挖掘。大约 20 个 NBA 球队使用了 IBM 公司开发的数据挖掘应用软件 Advanced Scout 系统来优化他们的战术组合。

例如，奥兰多魔术队教练曾利用 Scout 系统对队员进行不同的布阵安排，在与迈阿密热队的比赛中找到了获胜的机会。

系统分析显示，奥兰多魔术队先发阵容中的两个后卫安佛尼·哈德卫(Anfernee Hardaway)和伯兰·绍(Brian Shaw)在前两场中被评为－17 分，即这两个队员在场上，本队输掉的分数比得到的分数多 17 分。然而，当安佛尼·哈德卫与替补后卫达利尔·阿姆斯创(Darrell Armstrong)组合时，奥兰多魔术队得分为正 14 分。

在下一场中，奥兰多魔术队增加了达利尔·阿姆斯创的上场时间。这种方法果然见效：达利尔·阿姆斯创得到了 21 分，安佛尼·哈德卫得到了 42 分，奥兰多魔术队以 88∶79 获胜。奥兰多魔术队在第四场让达利尔·阿姆斯创进入先发阵容，再一次打败了迈阿密热队。在第五场比赛中，这个靠数据挖掘支持的阵容没能拖住迈阿密热队，但 Advanced Scout 系统毕竟帮助奥兰多魔术队赢得了打满 5 场，直到最后才决出胜负的机会。

Advanced Scout 系统是一个数据分析工具，教练可以用便携式计算机在家里或在路上挖掘存储在 NBA 中心的服务器上的数据。每一场比赛的事件都按得分、助攻、失误等进行统计分类。时间标记让教练非常容易地通过搜索 NBA 比赛的录像来理解统计发现的含义。例如，教练通过 Advanced Scout 系统发现本队的球员在与对方一个球星对抗时有犯规记录，他可以在对方球星与这个队员“头碰头”的瞬间分解双方接触的动作，进而设计合理的防守策略。

Advanced Scout 系统的开发人因德帕尔·布罕德瑞，在 IBM 的 Thomas·Watson 研究中心当研究员时，演示了一个技术新手应该如何使用数据挖掘。因德帕尔·布罕德瑞说：“教练们可以完全没有统计学的培训，但他们可以利用数据挖掘制定策略。”与此同时，另一个正式的体育联盟——国家曲棍球联盟，正在开发自己的数据挖掘应用 NHL-ICE，该联盟与 IBM 建立了一个技术型的合资公司，推出了一个电子实时的比赛计分和统计系统。在原理上是一个与 Advanced Scout 系统相似的数据挖掘应用，可以让教练、广播员、新闻记者及球迷挖掘 NHL 的统计。当他们访问 NHL 的 Web 站点时，球迷能够使用该系统循环观看联盟的比赛，同时广播员和新闻记者可以挖掘统计数据，找花边新闻，为实况评述添油加醋。

2. 超市中的数据挖掘

Safeway 是英国的第三大连锁超市，年销售额超过一百亿美元，提供的服务种类达三四十种。该超市的首席信息官 CIO 迈克·温曲指出，该公司必须要采用不同的方式来取得竞争上的优势。“运用传统的方法—降低价位、扩充店面以及增加商品种类，若想在竞争中取胜已经越来越困难了”。如何能在竞争中立于不败之地？温曲先生的说法是：“必须以客户为导向，而非以产品和商家为导向。这意味着必须更了解每一位客户的需求。为了达到这个目标，必须了解六百万客户所做的每一笔交易以及这些交易彼此之间的关联性。”换句话说，Safeway 想要知道哪些类型的客户买了哪些类型的产品以及购买的频率，用来建立“以个人为导向的市场”。

Safeway 首先根据客户的相关资料，将客户分为 150 类；再用关联技术来比较这些资料集合(包括交易资料以及产品资料)；然后列出产品相关度的清单(例如，“在购买烤肉炭的客

户中，75%的人也会购买打火机燃料”)；最后，再对商品的利润进行细分。例如，Safeway 发现某一种乳酪产品虽然销售额排名较靠后，在第 209 位，可是有 25%消费额最高的客户都常常买这种乳酪，这些客户是 Safeway 最不想得罪的客户。因此，这种产品是相当重要的。同时，Safeway 也发现，在 28 种品牌的橘子汁中，有 8 种特别受消费者欢迎。因此，该公司重新安排货架的摆放，使橘子汁的销量能够大幅增加。“我可以举出数百种与客户购买行为有关的例子，”温曲先生指出，“这些信息实在是无价之宝。”

采用数据挖掘技术，在 Safeway 知道客户每次采购时会买哪些产品后，就可以找出长期的经常性购买行为；再将这些资料与主数据库的人口统计资料结合在一起，营销部门就可以根据每个家庭在哪个季节倾向于购买哪些产品的特性发出邮件。根据这些信息该超市在一年内曾发了 1200 万封有针对性的邮件，对超市销售量的增长起了很重要的作用。

3. 站点访问量分析中的数据挖掘

美国亚特兰大的 AutoTrader. com 是世界上最大的汽车网站，网站上提供非常丰富的二手汽车及其他交通工具信息。每天有许多用户访问该网站，寻求有用的信息。

由于决策者需要从多角度、多层次来对客户访问网站的情况进行分析和管理，所以他们需要知道：什么样的客户访问这个网站、客户喜欢怎样的网站访问路径来获得所需信息、各个网站层次访问量如何、同一位客户访问网站的频率、客户经常重复进行怎样的购买行为、哪位老客户介绍来了新客户以及经介绍来的新客户和不是经介绍来的新客户购买习惯有什么不同，等等。最后，AutoTrader. com 决定用相关的分析和数据挖掘工具对用户的网络点击率进行分析，从而决定自己是否需要根据客户的不同喜好开设特定服务区。

AutoTrader. com 的数据存放在有 4 个处理器的 Sun Microsystems 4000 服务器上，选用了 SAS(Statistical Analysis System)的分析和数据挖掘软件，因为它们具有应用开发、信息和图形展现、Web 发布及 SAS/SPSS 等方面集成的能力，使得 AutoTrader. com 可对网站下一年度的访问流量进行预测。另外，由于它们可以很好地支持大数据量，AutoTrader. com 将不会为逐渐增长的数据量而担心。

不仅如此，AutoTrader. com 还定制了应用系统，每天凌晨两点应用系统访问日志文件中的数据，对数据自动解压与分析，自动生成包含访问统计量和图表的网页，并在第二天早晨自动送到决策人员的计算机上。这样，可以对这些报表进行操作，以报表或 3D 图表的形式进行浏览与观察。

4. 通过数据挖掘进行个性化服务

美国的 Big Sam’s Clothing 公司曾经开发了一个网站来补充商品目录。在 Big Sam’s 公司第一次将网站上线时，并没有什么个性化的内容，网站只是商品目录美观有效的在线翻版，但是没有利用 Web 增加销售机会。

后来，Big Sam’s 公司利用数据挖掘技术迅速提高了网络销售量。

首先，Big Sam’s 公司使用聚类的方法来发现哪些商品自然的分在同一组中。有时一些聚类是十分明显的，如衬衫和短裤；一些聚类可能是令人惊奇的，如关于沙漠探险的书和医疗工具包。当顾客购买其中的一种商品时，这些聚类用来向顾客提出购买另一种商品的建议。

然后，Big Sam's 公司又进行客户分析来识别那些经常对添加在商品目录中的新商品感兴趣的客户。Big Sam's 公司指引客户购买那些挑选出来的产品不仅仅带来销售的增加，而且巩固了客户关系。调查显示 Big Sam's 公司被看做是一个衣物和装饰品方面可信赖的顾问。

为了扩大影响，Big Sam's 公司还利用一个应用程序向客户发送 E-mail，通过 E-mail 向客户提供由数据挖掘模型预测的客户感兴趣的新产品信息。

个性化销售的努力为 Big Sam's 公司带来了盈利：它在重复销售、每一客户的平均销售量和销售的平均范围等方面有了较大的提高。

5. “体育精品”体育用品公司

“体育精品”体育用品公司总部在悉尼，在 7 个国家设有分店。

为了增加销售量，负责销售的副总裁决定通过奖励来促销，奖励销售额最高的地区和产品销售最多的单位。为此，这位副总裁要求首席信息官写出两份报告，但是销售数据存储在不同地区的不同类型的数据库中，不但数据的格式不同，而且不同地区营业额用所用货币单位也不同。首席信息官先用 IBM Visual Warehouser 数据仓库工具将这些数据集中，并进行处理。完成了副总裁要求的两份报告：按地区的销售额和按产品的销售额。

首席信息官向副总裁建议，可以进一步挖掘其他信息。如购买山地车的顾客最可能再购买其他哪些产品，购买气瓶的顾客 1 年内回来充气多少次。得到的答案如下。

(1) 购买山地车的顾客再购买头盔的可能性为 92%；再购买手套的可能性为 62%；再购买新款铃铛的可能性为 23%；再购买速度计的可能性为 13%。

通过上述数据，决定对购买山地车的顾客引导他们再购买上述产品；还可以对他们宣传骑车安全问题，提高购买反光罩、车灯和后视镜等产品的销售；也可以向顾客进行饮料瓶、个人音响等其他产品的捆绑销售。

(2) 购买气瓶的顾客一年内回来充气 1 次的有 12%；回来充气 2 次的有 8%；回来充气 2 次以上的只有 7%。

针对上述数据，有两种决策：放弃充气业务或进行更大力度的促销策略。

决定采取第二种决策，具体促销策略是，给两次以上充气的顾客优惠 25% 折扣，或实施新的刺激销售方法，即在春季给购买气瓶的顾客邮寄信函提醒他们回来充气，在停车场建立更多的便利充气站以及顾客每一次充气都发折扣优惠券等。

一个月后，季度的营业额上升 34%，收入上涨 32%。每辆山地车交易的平均销售收入增加了 29%，山地车与头盔一起购买成了时尚，手套的销售额上升了 15%，山地车附件的销售额上升了 51%。捆绑销售得到普及，气瓶充气的销售开始上升。

1.4 常用数据挖掘工具

目前，数据挖掘在企业经营管理、政府行政管理的决策支持及科学研究等领域获得了广泛的应用，许多公司已经推出了专门的数据挖掘工具，如 IBM 公司的 Intelligent Miner，Thinking Machines 公司的 Darwin，NeoVista Solution 公司的 Decision Series，Angoss

的 Knowledge Seeker 等;另外,很多数据库管理系统或者统计软件也增加了支持数据挖掘的功能,如 SQL Server 2005、Oracle、SPSS/Clementine 及 SAS/Enterprise Miner 等。由于不同的工具具有不同的优势和不足,因此,要想真正做好数据挖掘,就要根据所选择的数据对象和需求,选择合适的数据挖掘工具。本节将介绍常用的几种数据挖掘工具及其应用。

1.4.1 数据挖掘工具的种类

1. 按使用方式分类

数据挖掘工具按使用方式分类,可以分成决策方案生成工具、商业分析工具和研究分析工具。

决策方案生成工具是针对某个特定行业或特定问题而开发的一类数据挖掘工具,如金融行业的欺诈检查工具、零售行业的客户流失分析工具等。

商业分析工具包括两种类型:一种是只为用户提供一个黑箱,用户只需要将需要分析的对象和相关的一些环境因素提供给工具,数据挖掘工具将自动给出数据挖掘的结果,其内部的一些复杂模型并不向用户展示,这种类型的数据挖掘工具适合管理人员使用;另一种是向用户展示数据挖掘模型,用户可以根据自己的需要选择数据挖掘模型或对数据挖掘模型进行适当的控制。例如,将决策树展示给用户,用户可以对决策树进行切片处理,这类工具主要为企业管理顾问或商业分析人员服务。

研究分析工具为用户提供了更大的数据挖掘应用空间,其用户主要是数据挖掘研究人员或商业分析人员。这些工具包含了一些数据挖掘研究领域的最新研究成果,如文本挖掘、Web 挖掘或图形、可视化工具等。

2. 按数据挖掘技术分类

数据挖掘工具按数据挖掘技术的种类,可以分成基于神经网络的工具,基于规则和决策树的工具,基于模糊逻辑的工具和综合性数据挖掘工具等。

(1) 基于神经网络的工具。该工具由于有非线性数据的快速建模能力,在实际应用中越来越流行,特别是在市场数据库的分析和建模方面应用比较广泛。

(2) 基于规则和决策树的工具。采用规则发现或决策树分类技术来发现数据模式和规则,其核心是某种归纳算法。这类工具常针对数据库的数据进行开发,生成规则和决策树,然后对数据进行分析和预测,其主要优点是规则和决策树都是可读的。

(3) 基于模糊逻辑的工具。其数据挖掘方法是应用模糊逻辑进行数据查询、排序等。该工具使用模糊概念和“最近”搜索技术的数据查询工具,它可以让用户指定目标,然后对数据库进行搜索,找出接近目标的所有记录,并对结果进行评估。

(4) 综合性数据挖掘工具。这类工具采用了多种数据挖掘方法,一般规模较大,适合对大型数据库的数据进行挖掘。综合性数据挖掘工具的数据挖掘能力很强,但价格昂贵,并且用户需要花很长的时间进行学习,才能掌握这类工具的应用。

3. 按应用范围分类

数据挖掘工具按应用范围,可以分成专用型数据挖掘工具和通用型数据挖掘工具。

(1) 专用型数据挖掘工具主要用于某一特定领域。如美国加州理工学院与日本的

Kayyad 设计的 SKICAT,能够对大规模的空间数据进行分析,识别遥远空间的星体。芬兰赫尔辛基大学研制的 TASA,能够采用特殊算法处理网络通信中的数据,对网络通信故障发出警报。由于专用型数据挖掘工具针对性较强,采用了一些特殊的算法对特定的数据集进行处理,数据挖掘的效率较高,挖掘出的知识可靠性也较高,但是应用范围受到限制。

(2) 通用型数据挖掘工具有 IBM 公司的 IM 智能挖掘器,这是一套包括了 Explorer、Diamond 和 Quest 在内的软件产品,可以用来提供高端数据挖掘的解决方案。其中的 Explorer 是一种用于聚类的神经网络工具,Diamond 是一种可视化数据挖掘软件产品,而 Quest 则提供了关联规则、分类规则、序列模式与相似序列等模式。

SPSS 公司的统计软件包 SPSS 在统计领域处于领先的位置,其中的线性回归分析结果和类似的数据挖掘工具对数据挖掘的结果是一致的,而这些挖掘工具采用的是传统统计方法。

Red Brick 系统公司的 Red Brick 数据挖掘工具是第一个将数据挖掘解决方案与数据库集成在一起的数据挖掘软件。在与数据库的链接中减少了传统数据挖掘中需要的大量数据准备时间,并且提供了扩展的 SQL 语言,用户可以使用 SQL 语言建立、存取和访问数据仓库中的模型。

1.4.2 评价数据挖掘工具优劣的指标

在数据挖掘技术日益发展的同时,许多数据挖掘的商业软件工具也逐渐问世。评价一个数据挖掘工具,主要从 5 个方面来考虑:可产生的模式种类的数量;解决复杂问题的能力;易操作性;数据存取能力;与其他产品的接口、噪声数据的处理及挖掘工具的鲁棒性。

评价数据挖掘工具优劣的指标如下。

(1) 数据准备。包括数据净化、描述、变换和抽样的能力。

(2) 数据访问。即访问不同数据源的能力。

(3) 算法建模。数据挖掘寻找的知识类型多种多样,有关联规则、分类/预测、聚类规则等模型,因此,优秀的挖掘工具应当包含多种数据挖掘算法,以处理不同的需求;同时,算法的稳定性、收敛性以及对噪声的敏感程度等也是重要指标。

(4) 模型的评价和解释。数据挖掘工具经过对数据的分析建立模型,要求工具能够提供多样的、易于理解的方式,如模型的性能参数、图表表示方法等,对模型进行评价和解释。

(5) 用户界面。数据挖掘工具经过对数据的分析建立模型,部分工具还提供了数据挖掘应用编程接口(application programming interface,API),是为专业用户而配置的。相比之下,图形用户接口(graphics user interface,GUI)可以简化建模的过程,方便普通用户。能否满足不同类型用户的需求,也是评价工具的重要指标。

1.4.3 常用数据挖掘工具

数据挖掘工具种类繁多,以下介绍几种常用的数据挖掘工具。

1. SPSS

SPSS(Statistical Package for the Social Science,社会科学统计软件包)是一种集成化

的计算机数据处理应用软件，主界面如图 1-2 所示。1968 年，美国斯坦福大学 H. Nie 等 3 位大学生开发了最早的 SPSS 统计软件，并于 1975 年在芝加哥成立了 SPSS 公司，广泛应用于通信、医疗、银行、证券、保险、制造、市场研究、科研、教育等多个领域和行业。

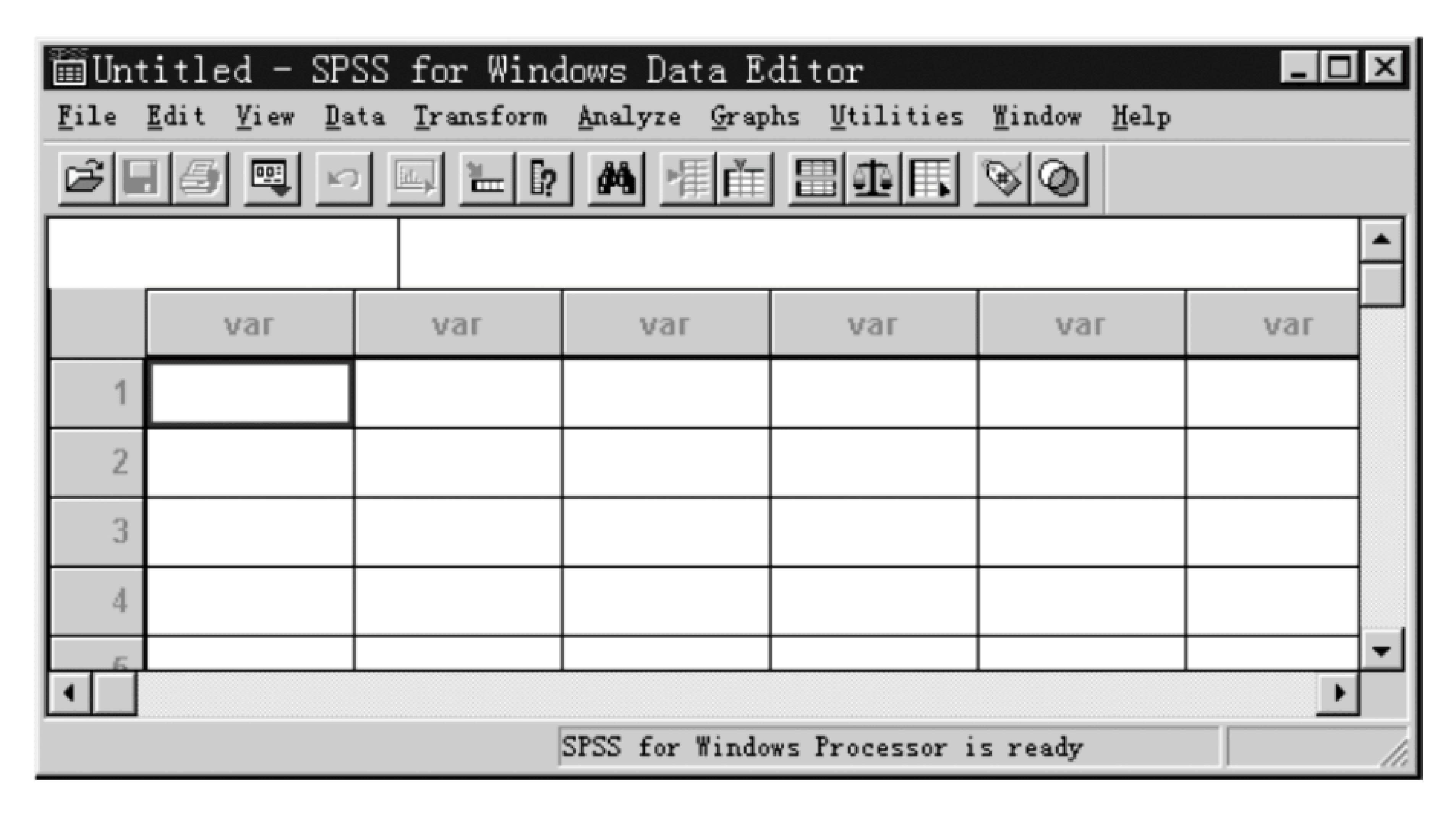


图 1-2　SPSS 界面

目前，世界上最著名的数据分析软件是 SAS 和 SPSS。SAS 是为专业统计分析人员设计的，具有功能强大，灵活多样的特点，为专业人士所喜爱。而 SPSS 是为广大的非专业人士设计，它操作简便，好学易懂，简单实用，因而很受非专业人士的青睐。此外，比起 SAS 软件来，SPSS 主要针对社会科学研究领域开发，因而更适合应用于教育科学研究，是国外教育科研人员必备的科研工具。1988 年，中国高教学会首次推广了这种软件，从此成为国内教育科研人员最常用的工具。

SPSS 软件的特点如下。

(1) 集数据录入、资料编辑、数据管理、统计分析、报表制作、图形绘制为一体。从理论上说，只要计算机硬盘和内存足够大，SPSS 可以处理任意大小的数据文件，无论文件中包含多少个变量，也不论数据中包含多少个案例。

(2) 统计功能囊括了几乎全部常规的统计方法，并能在屏幕(或打印机)上显示(打印)如正态分布图、直方图、散点图等各种统计图表。从某种意义上讲，SPSS 软件还可以帮助数学功底不够的使用者学习运用现代统计技术。使用者仅需要关心某个问题应该采用何种统计方法，并初步掌握对计算结果的解释，而不需要了解其具体运算过程，可以在使用手册的帮助下定量分析数据。

(3) 界面友好，操作简单。SPSS for Windows 界面完全是菜单式，一般稍有统计基础的人经过简单培训即可用 SPSS 做简单的数据分析。

2. SAS

SAS 是由美国北卡罗来纳州立大学于 1966 年开发的统计分析软件。1976 年 SAS 软件研究所(SAS Institute Inc.)成立，开始进行 SAS 系统的维护、开发、销售和培训工作。经过多年来的完善和发展，SAS 系统在国际上已被誉为统计分析的标准软件，在各个领域得到广泛应用。

SAS 是一个模块化、集成化的大型应用软件系统，主界面如图 1-3 所示。它由数十个专用模块构成，功能包括数据访问、数据储存及管理、应用开发、图形处理、数据分析、报告编制、运筹学方法、计量经济学与预测等。SAS 系统基本上可以分为四大部分：SAS 数据库部分、SAS 分析核心、SAS 开发呈现工具、SAS 对分布处理模式的支持及其数据仓库设计，分别完成以数据为中心的四大任务：数据访问、数据管理、数据呈现、数据分析。

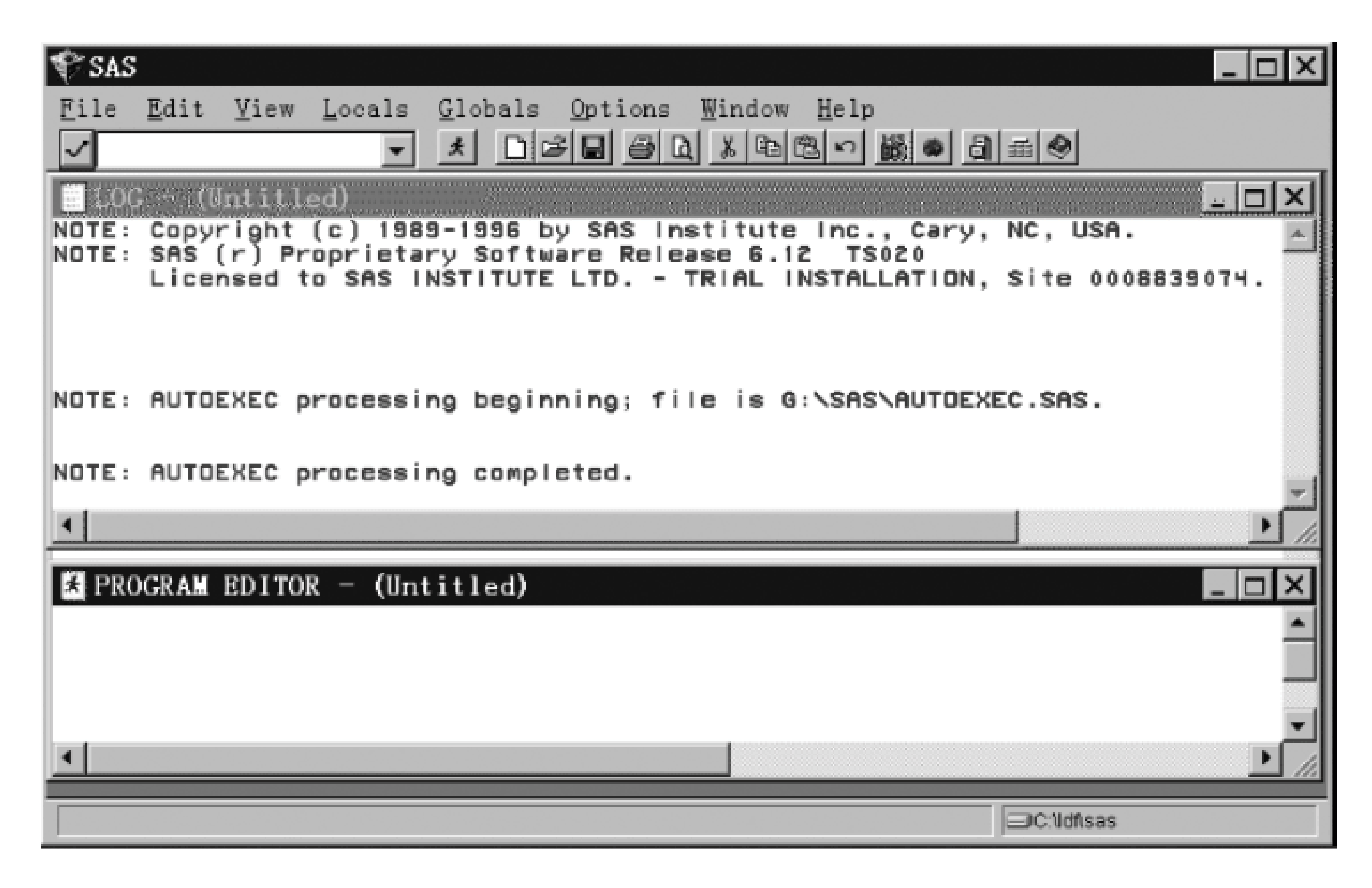


图 1-3　SAS 界面

SAS 软件具有以下特点。

(1) 功能强大，统计方法齐、全、新。SAS 提供了从基本统计数的计算到各种实验设计的方差分析、相关回归分析、多参数分析等多种统计分析过程，几乎囊括了所有最新分析方法，其分析技术先进、可靠。分析方法的实现通过过程调用完成。许多过程同时提供了多种算法和选项。

(2) 使用简便，操作灵活。SAS 编程语句简洁，短小，通常只需很小的几句语句即可完成一些复杂的运算，得到满意的结果。结果输出以简明的英文给出提示，统计术语规范易懂。

(3) 提供联机帮助功能。使用过程中按下功能键 F1，可随时获得帮助信息，得到简明的操作指导。SAS 把数据存取、管理、分析和展现有机地融为一体。

3. SQL Sever 2005

SQL Server 是一个全面的、集成的、端到端的数据解决方案，它为组织中的用户提供了一个更安全可靠和更高效的平台，主要用于企业数据和 BI 应用。SQL Server 2005 为 IT 专家和信息工作者带来了功能强大的数据挖掘分析工具，同时降低了在从移动设备到企业数据系统的多平台上创建、部署、管理和使用企业数据和分析应用程序的复杂性。

SQL Server 2005 也包含了多个能显著提高开发者能力的新技术。从支持.NET 框架到和 Visual Studio 的紧密集成，这些新特性使开发人员能够以更低的成本，更容易地创建安全、强大的数据库应用程序。

Microsoft SQL Server 2005 Data Mining(数据挖掘)属于商务智能技术，可帮助构建复杂的分析模型，并使其与业务操作相集成。Microsoft SQL Server 2005 分析服务中构建了一个新的易于使用的、容易扩展的、方便访问的、非常灵活的平台，如图 1-4 所示。对于以前

从未考虑过采用数据挖掘的组织机构,这无疑是个非常容易接受的解决方案。



图 1-4　Microsoft SQL Server 2005 数据挖掘平台界面

4. Weka

Weka(Waikato Environment for Knowledge Analysis,怀卡托智能分析环境),是一个开放源码的数据挖掘软件,主界面如图 1-5 所示。Weka 也是新西兰独有的一种鸟名,而 Weka 的主要开发者来自新西兰的 Waikato 大学。数据挖掘用户可通过 Weka 集成的大量算法,执行数据预处理、分类、回归、聚类、关联规则、数据可视化等任务。而开发者可使用 Java 语言,在 Weka 架构上开发出更多的数据挖掘算法。使用 Weka 可以轻松地进行数据预处理和在数据集上运用数据挖掘算法。

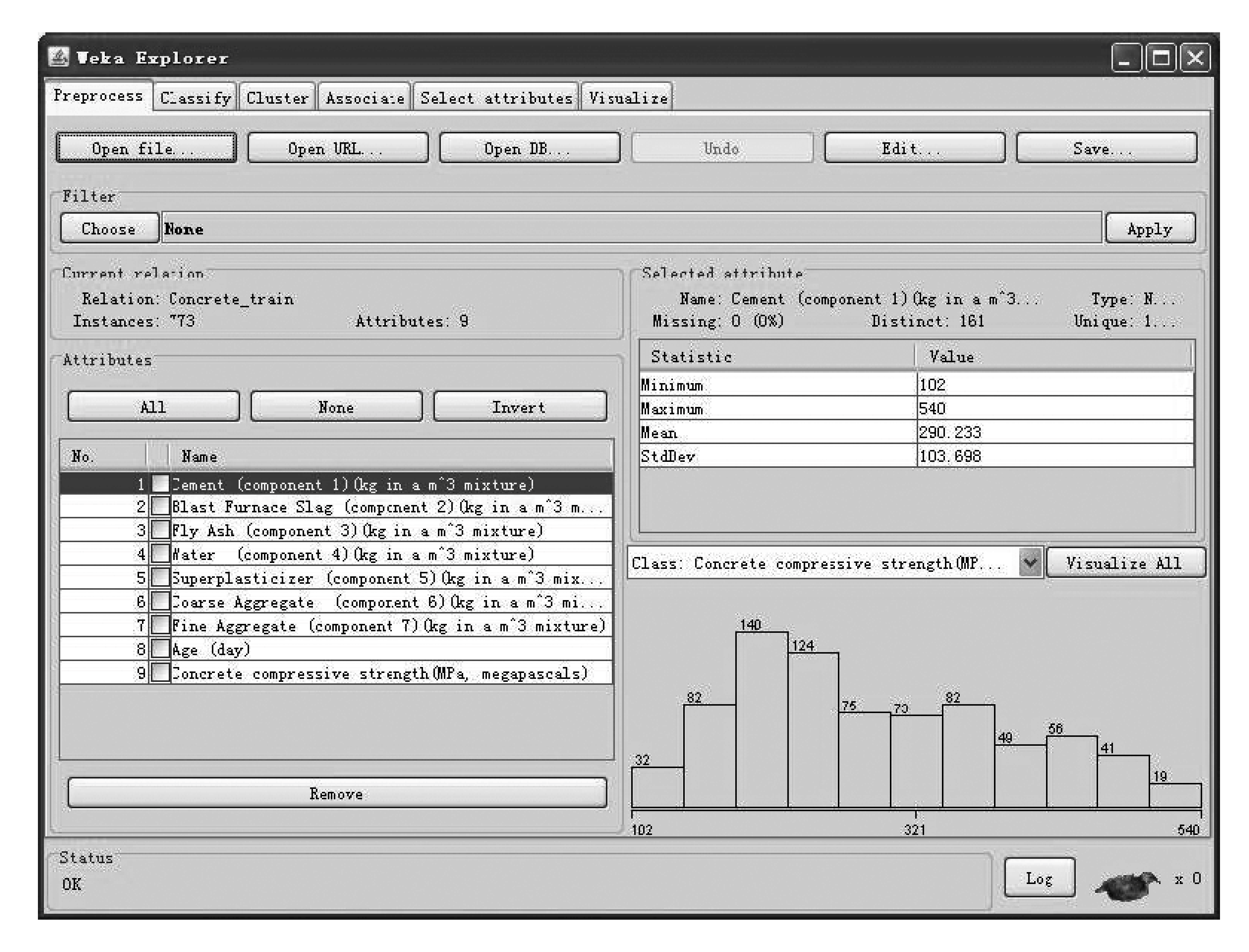


图 1-5　Weka 界面

5. MATLAB

MATLAB是矩阵实验室(Matrix Laboratory)的简称,是美国 MathWorks 公司出品的商业数学软件,是用于算法开发、数据可视化、数据分析以及数值计算的高级计算语言和交互式环境,主要包括 MATLAB 和 Simulink 两大部分,主界面如图 1-6 所示。

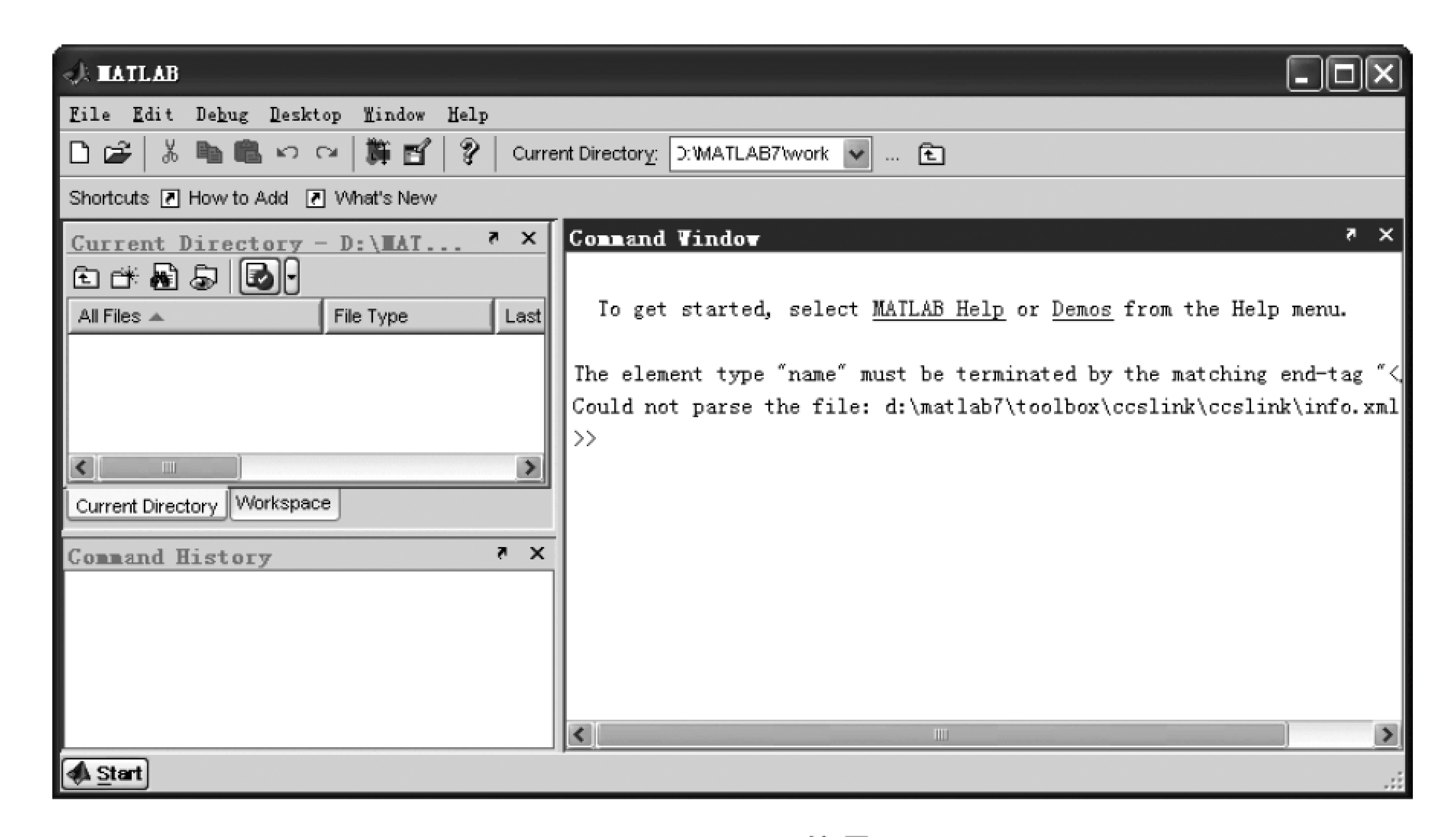


图 1-6 MATLAB 的界面

MATLAB 的基本数据单位是矩阵,它的指令表达式与数学、工程中常用的形式十分相似,故用 MATLAB 来解算问题要比用 C,FORTRAN 等语言完成相同的事情简捷得多,并且 MATLAB 也吸收了 Maple 等软件的优点,使 MATLAB 成为一个强大的数学软件。在新的版本中也加入了对 C、FORTRAN、C++ 、Java 的支持,可以直接调用,用户也可以将自己编写的实用程序导入到 MATLAB 函数库中,方便自己以后调用。

MATLAB 的应用范围非常广,包括信号和图像处理、通信、控制系统设计、测试和测量、财务建模和分析以及计算生物学等众多应用领域。附加的工具箱(单独提供的专用 MATLAB 函数集)扩展了 MATLAB 环境,以解决这些应用领域内特定类型的问题。

小结

本章重点介绍了数据挖掘技术的相关概念,包括数据挖掘的由来、数据挖掘的定义、数据挖掘的功能和数据挖掘的常用技术。然后,介绍了数据仓库的基本概念,包括数据仓库的产生与发展,数据仓库的定义,数据仓库和数据挖掘的关系。接着,通过有说服力的数据挖掘应用实例,使读者进一步体会到应用数据挖掘技术的必要性。最后,介绍了数据挖掘常用工具及其特点,以方便读者选择适合的数据挖掘工具。

习题 1

1. 数据挖掘技术涉及哪些技术领域?
2. 数据挖掘的源数据是否必须是数据仓库的数据?可以有哪些来源?

3．数据挖掘的具体功能有哪些？
4．数据挖掘技术主要包含哪几种？
5．数据挖掘的过程包括哪些步骤，每一步具体包括哪些内容？
6．数据挖掘可以应用在哪些领域？
7．数据库与数据仓库的本质区别是什么？
8．举例说明数据挖掘与数据仓库的关系。
9．举例说明数据挖掘从数据仓库中挖掘的信息有哪些？
10．搜索数据挖掘的应用实例。
11．数据挖掘工具的主要指标有哪些？
12．常用数据挖掘工具有哪些？各有什么特点？

第 2 章　数 据 仓 库

数据仓库(data warehouse,DW)是一种环境,不是一种产品。它包括电子邮件文档、语音文档、CD-ROM、多媒体信息以及还未考虑到的数据。数据仓库中的数据并非是最新的、专有的,而是来源于其他的数据库。数据仓库的建立并不是要取代原有的数据库,而是建立在一个较全面、完善的信息应用的基础上,用于支持高层决策分析。

2.1　数据仓库的基本概念

数据仓库是一个环境,而不是一件产品,提供用户用于决策支持的当前的和历史的数据,这些数据在传统的操作型数据库中很难或不能得到。

传统的数据库系统面向以事务处理为主的联机处理系统的应用,不能满足决策支持系统(decision sustain system,DSS)的分析要求。事务处理和分析处理具有不相同的性质,因而两者对数据也有着不同的要求。W. H. Inmon 在其《建立数据仓库》一书中,列出了操作型数据与分析型数据之间的区别,如表 2-1 所示。

表 2-1　操作型数据与分析型数据的区别

操作型数据	分析型数据
细节的	综合的或提炼的
在存取瞬间是准确的	代表过去的数据
可更新	不更新
操作需求事先可知道	操作需求事先不知道
生命周期符合软件生命周期	完全不同的生命周期
对性能要求高	对性能要求宽松
一个时刻操作一单元	一个时刻操作一集合
事务驱动	分析驱动
面向应用	面向分析
一次操作数据量小	一次操作数据量大
支持日常操作	支持管理需求

上述操作型数据与分析型数据之间的差别从根本上体现了事务处理与分析处理的差异。传统的数据库系统主要用于企业的日常事务处理工作,存放在数据库中的数据也就大体符合操作型数据的特点。而为适应数据分析处理要求而产生的数据仓库中所存放的数据就应该是分析型的数据。表 2-1 中所列出的分析型数据的特点可以概括为 4 点,也就是数据仓库数据的 4 个基本特征:

(1) 数据仓库的数据是面向主题的;

(2) 数据仓库的数据是集成的;

(3) 数据仓库的数据是不可更新的;

(4) 数据仓库的数据是随时间不断变化的。

可以概括地定义,数据仓库就是一个用以更好地支持企业或组织的决策分析处理的、面向主题的、集成的、不可更新的、随时间不断变化的数据集合。下面着重来讨论数据仓库数据的 4 个基本特征。

1. 数据仓库的数据是面向主题的

传统数据库是面向应用的,为每个单独的应用程序组织数据。数据仓库中的数据是面向主题进行组织的;面向主题性是数据仓库中数据组织的基本原则,数据仓库中的所有数据都是围绕着某一主题组织、展开的。

(1) 主题的概念。主题是一个抽象的概念,是在较高层次上将企业信息系统中的数据综合、归类并进行分析利用的抽象,在逻辑关系上,它对应企业中某一宏观分析领域所涉及的分析对象。面向主题的数据组织方式,就是在较高层次上对分析对象数据的一个完整、一致的描述,能完整、统一地刻画各个分析对象所涉及的企业各项数据,以及数据之间的联系。

从信息管理的角度看,主题就是在一个较高的管理层次上对信息系统中的数据按照某一具体的管理对象进行综合、归类所形成的分析对象。

从数据组织的角度看,主题就是一些数据集合,这些数据集合对分析对象进行了比较完整的、一致的数据描述,这种描述不仅涉及数据自身,还涉及数据之间的关系。

数据仓库的创建、使用都是围绕主题实现的,因此,必须了解如何按照决策分析来抽取主题,所抽取的主题应该包含哪些数据内容,这些数据内容应该如何组织。

例如,在企业销售管理中的管理人员,所关心的是,本企业哪些产品销售量大、利润高,哪些客户采购的产品数量多,竞争对手的哪些产品对本企业产品构成威胁,根据这些管理决策的分析对象,就可以抽取出“产品”、“客户”等主题。

(2) 主题划分的原则。在划分主题时必须保证每个主题的独立性,每一个主题要具有独立的内涵,明确的界限。在划分主题时,需要保证对主题进行分析时所需的数据都可以在此主题内找到,保证主题的完备性。

确定主题以后,需要确定主题应该包含的数据,此时应该注意不能将围绕主题的数据与业务处理系统的数据相混淆。

在主题的数据组织中应该注意,不同的主题之间可能出现相互重叠的信息,这种主题间重叠是逻辑的,而不是同一数据内容的物理存储重复。例如,“客户”主题与“产品”主题在产品购买信息方面有相互重叠的信息,是源于客户和产品都有关的销售业务处理系统。

需要指出一点,目前数据仓库仍是采用关系数据库技术来实现的,也就是说数据仓库的数据最终也表现为关系。

(3) 主题划分的实例。为了更好地理解主题与面向主题的概念,说明面向主题的数据组织与传统的面向应用的数据组织方式的不同。例如:一家采用“会员制”经营方式的商场,按业务已建立起销售、采购、库存管理以及人事管理子系统。

① 面向应用的数据组织。该商场按照不同的应用建立了各自的数据库模式,各子系统建立数据库情况如表 2-2 所示。

② 面向主题的数据组织。按照面向主题的方式,数据的组织应该分为两个步骤:抽取主题以及确定每个主题所应包含的数据内容。

表 2-2 各子系统建立数据库情况

子 系 统	数据库名称	数 据 字 段
销售子系统	顾客	顾客号,姓名,性别,年龄,文化程度,地址,电话
	销售	员工号,顾客号,商品号,数量,单价,日期
采购子系统	订单	订单号,供应商号,总金额,日期
	订单细则	订单号,商品号,类别,单价,数量
	供应商	供应商号,供应商名,地址,电话
库存管理子系统	领料单	领料单号,领料人,商品号,数量,日期
	进料单	进料单号,订单号,进料人,收料人,日期
	库存	商品号,库房号,库存量,日期
	库房	库房号,仓库管理员,地点,库存商品描述
人事管理子系统	员工	员工号,姓名,性别,年龄,文化程度,部门号
	部门	部门号,部门名称,部门主管,电话

主题的抽取是按照分析的要求来确定的。概括各种分析领域的分析对象,可以综合得到主题。上例的主题应包括供应商、商品、顾客等。每个主题有着各自独立的逻辑内涵,对应了一个分析对象。按照面向主题的数据组织如表 2-3 所示。

表 2-3 面向主题的数据组织

主 题	信 息 类	数 据 字 段
商品	商品固有信息	商品号,商品名,类别,颜色
	商品采购信息	商品号,供应商号,供应价,供应日期,供应量
	商品销售信息	商品号,顾客号,售价,销售日期,销售量
	商品库存信息	商品号,库房号,库存量,日期
供应商	供应商固有信息	供应商号,供应商名,地址,电话
	供应商品信息	供应商号,商品号,供应价,供应日期,供应量
顾客	顾客固有信息	顾客号,顾客名,性别,年龄,文化程度,住址,电话
	顾客购物信息	顾客号,商品号,售价,购买日期,购买量

以“商品”主题为例,关于商品的各种信息已综合在“商品”主题中,包含了商品的固有信息(商品名称、商品类别、型号和颜色等商品的描述信息)和商品的流动信息(商品采购信息、商品销售信息及商品库存信息)。丢弃了原来不必要的、不适于分析的信息,如有关订单、领料单等信息;并将原来分散在不同数据库中的商品信息集成在一起,形成了关于商品一致的信息集合。

不同的主题之间有重叠的内容,例如“商品”主题的商品销售信息同“顾客”主题的顾客购物信息有些数据字段是相同的,这表现了“顾客”和“商品”这两个主题之间的联系;同样,“商品”主题中有些信息同“供应商”主题中的某些信息相同,这表现的是“商品”和“供应商”

之间的联系。但主题间的重叠并不是两两重叠，如“供应商”和“顾客”主题间是没有重叠内容的，这表现了“供应商”和“顾客”之间不发生直接的联系，而是通过“商品”主题来表现它们之间的间接联系。“商品”、“顾客”和“供应商”这 3 个主题间的关系如图 2-1 所示。

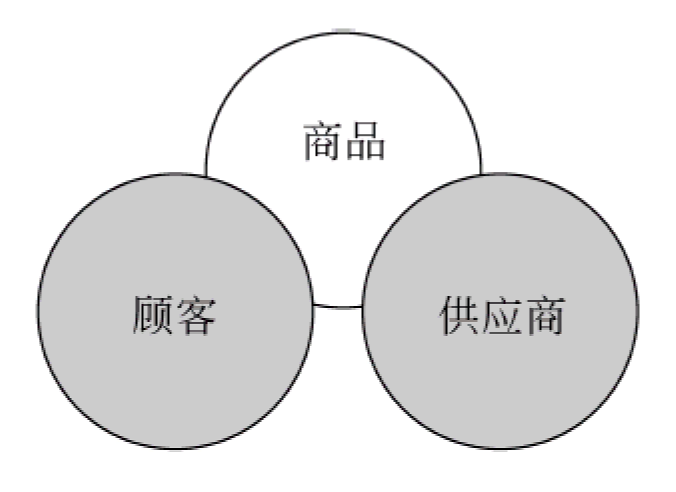


图 2-1　主题间的重叠关系

一个主题可以划分成多个表，基于一个主题的所有表都含有一个称为公共码键的属性作为其主码的一部分。公共码键将各个表统一联系起来，体现它们是属于一个主题的。如“商品”主题的“商品号”就是公共码键。

2. 数据仓库的数据是集成的

数据仓库的数据是从原有分散的数据库、数据文件和数据段中抽取来的，数据来源可能既有内部数据又有外部数据。面向应用的数据与面向主题的数据之间差别很大。因此，在数据进入数据仓库之前，必然要经过转换、统一与综合。这一步是数据仓库建设中最关键、最复杂的一步。

（1）统一源数据

面向应用的设计人员经过多年的不同设计会有许多不同的表示方法，不同的设计人员也会采用不同的表示，没有一个统一的标准。例如，表示性别，可以用“男”、“女”，也可以用 Y、Z 等。因此，当数据进入数据仓库时，必须消除面向应用的数据的不一致性，将源数据统一。

统一源数据的内容如下：

① 命名规则；

② 编码；

③ 数据特征；

④ 度量单位。

（2）综合和计算。许多情况下，在从原有数据库抽取数据生成数据仓库时，并不仅仅是原封不动地“复制”过来，而需要进行综合和计算。例如，上例中销售子系统的“销售”数据库按照顾客每一次的购买作为一条记录。而在数据仓库中顾客主题的“顾客购物信息”中，可以按天、周、月等组织数据。很明显，这种情况就需要对数据进行计算和综合。

数据仓库中的数据综合工作可以在从原有数据库抽取数据时生成，但许多是在数据仓库内部生成的，即进入数据仓库以后进行综合生成的。

3. 数据仓库的数据是不可更新的

从操作型系统中提取的数据和从外部数据源中提取的数据，在数据仓库中被转换、综合并存储。数据仓库的数据主要供企业决策分析之用，不是用来进行日常操作，一般只保存过去的数据，而且不是随着源数据的变化实时更新，数据仓库中的数据一般不再修改。所涉及的数据操作主要是数据查询，只定期进行数据加载、数据追加，一般情况下并不进行修改操作。

由于数据仓库的数据是不可更新的，因此，也称其为具有非易失性或非易变性。这种不可更新性可以支持不同的用户在不同的时间查询相同的问题时获得相同的结果。

4. 数据仓库的数据是随时间不断变化的

（1）数据仓库的数据随时间变化。数据仓库中的数据不可更新是针对应用来说的，即

数据仓库的用户进行分析处理时是不进行数据更新操作的。但并不是说,在从数据集成输入数据仓库开始到最终被删除的整个数据生存周期中,所有的数据仓库数据都是永远不变的。

数据仓库的数据随时间的不断变化主要体现在数据仓库随时间变化不断增加新的数据内容,删去旧的数据内容,数据仓库中所包含的综合数据经常按照时间段进行综合,隔一定的时间片进行抽样等重新综合。

因此,数据仓库数据的码键都包含时间项,以标明数据的历史时期。

(2)数据仓库的数据追加。如何定期向数据仓库追加数据也是一个十分重要的技术。数据仓库的数据加载完成后,再向数据仓库输入数据的过程称为数据追加。数据追加的内容仅限于上次向数据仓库输入后原来数据库中变化了的数据。因此,要完成数据追加,必需能够明确哪些数据是在上一次追加过程之后新生成的,这项工作称为变化数据的捕捉。捕捉变化数据的常用方法如下。

① 时标方法。如果数据含有时标,那么只需根据时标即可判断哪些数据是上次追加后变化了。但许多数据库中的数据并不含有时标。

② DELTA 文件。DELTA 文件是由应用生成的,记录了应用改变的所有内容。利用 DELTA 文件,效率比较高,避免了扫描整个数据库,但生成 DELTA 文件的应用并不普遍。

③ 前后映像文件的方法。将抽取数据到数据仓库时的数据库单独“保存”(称之为“快照”),在下一次抽取数据库数据时,对数据库再作一次快照。比较前后两幅快照的不同,从而确定实现数据仓库追加的数据。这种方法需占用大量资源。

④ 日志文件。日志文件是数据库的固有机制,不会额外增加工作量和占用系统资源。提取数据只局限于日志文件,不用扫描整个数据库;当然,原来日志文件的格式是依据数据库系统的要求而确定的,它包含的数据对于数据仓库而言可能有许多冗余,如对一个记录的多次更新,日志文件将全部变化过程都记录下来,而对于数据仓库,只需要最终结果。因此,在利用日志文件进行数据仓库的数据追加时,同样需要进行数据的转换、综合和鉴别等。

传统数据库在联机事物处理中取得了较大的成功,但在基于事物处理的数据库帮助决策分析时却产生了很大的困难。主要原因是传统数据库的处理方式和决策分析中的数据需求不相称,导致传统数据库无法支持决策分析活动。这些不相称主要体现在决策处理的响应较慢,决策数据需求得不到满足,决策数据操作不能满足用户的需求等。

数据仓库虽然是从数据库发展而来的,但是两者在许多方面存在着相当大的差异(见表 2-4)。从数据存储内容看,数据库只存放当前值,而数据仓库则存放历史值;数据库中数据的目标是面向业务操作人员的,为业务处理人员提供信息处理的支持,而数据仓库则是面向中高层管理人员的,为其提供决策支持。数据库内数据是动态变化的,只要有业务发生,数据就会被更新,而数据仓库则是静态的历史数据,只能定期添加、刷新。数据库中的数据结构比较复杂,有各种结构以适合业务处理系统的需要,而数据仓库中数据的结构则相对简单。数据库中数据访问频率较高,但访问量较少,而数据仓库的访问频率较低但访问量却远高于数据库的访问量。数据库在访问数据时要求响应速度快,其响应时间一般在几秒内,而数据仓库的响应时间则可长达数小时。

表 2-4　数据仓库与数据库对比

对比内容	数　据　库	数 据 仓 库
数据内容	当前值	历史的、存档的、归纳的、计算的数据
数据目标	面向业务操作程序，重复处理	面向主题域，分析应用
数据特性	动态变化，按字段更新	静态、不能直接更新，只能定时添加、刷新
数据结构	高度结构化、复杂，适合操作计算	简单、适合分析
使用频率	较高	中到低
数据访问量	每个事物只访问少量记录	有的事物可能需要访问大量记录
响应要求	以秒为单位	时间长

2.2　数据仓库的体系结构

数据仓库体系结构可用图 2-2 表示。由于数据库和数据仓库应用的出发点不同，数据仓库将独立于业务数据库系统，但是数据仓库又同业务数据库系统息息相关。也就是说数据仓库不是简单地对数据进行存储，而是对数据进行“再组织”。数据仓库的体系结构框架是影响数据仓库性能的关键因素之一，数据仓库的体系结构框架决定了数据加载、访问和传递的方式。在确定数据仓库结构时需要考虑最终用户和数据使用部门的数目、数据的多样性和数量、更新周期以及存储访问的速度。在数据仓库体系结构中应该设计三个独立的数据层次：信息获取层、信息存储层和信息传递层。信息获取层负责数据的收集、提取、净化和聚合，以及从外部数据源和业务处理系统中获取数据。这些数据应该是准确的，并且要被用于各个部门进行决策支持，因此需要有通用的含义。信息存储层是一个保存数据的区域，这些信息是在信息传递层次中可以得到的信息。支持集成传递所必需的性能要求之一就是

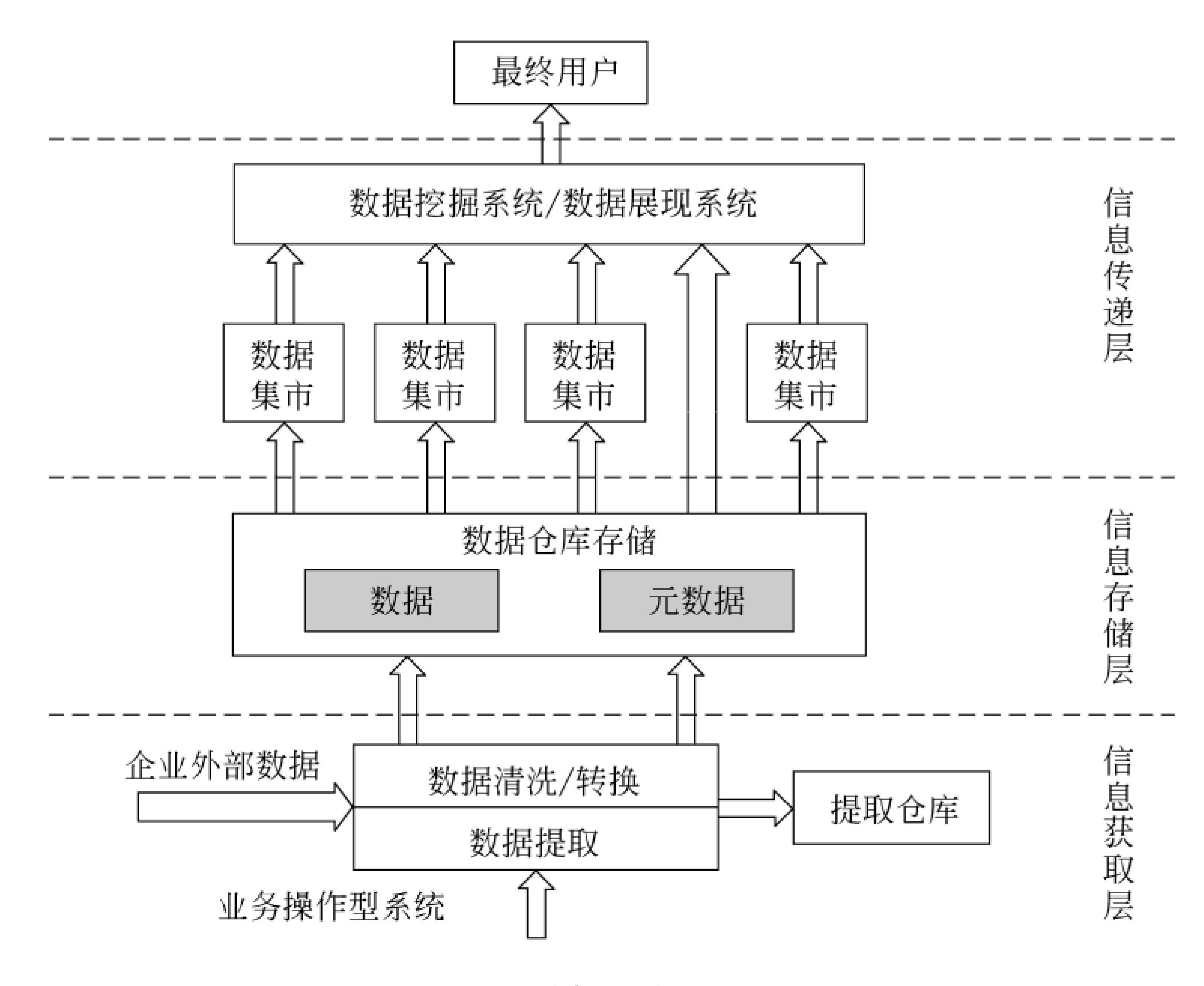


图 2-2　数据仓库体系结构

灵活性,在数据仓库体系结构中需要利用信息传递层来实现灵活性。信息传递层通过生成的报表和查询来提供数据需求。这是最终用户与数据仓库交流的层次,也是数据仓库与用户接触的地点。

数据仓库中的数据可分为多个级别。下例中的数据仓库的数据分为 4 个级别:早期细节级、当前细节级、轻度综合级和高度综合级。源数据经过综合后,首先进入当前细节级,并根据具体需要进行进一步的综合从而进入轻度综合级乃至高度综合级,老化的数据将进入早期细节级。数据仓库中存在着的不同综合级别,将其称之为粒度(granularity)。数据仓库的数据组织结构如图 2-3 所示。

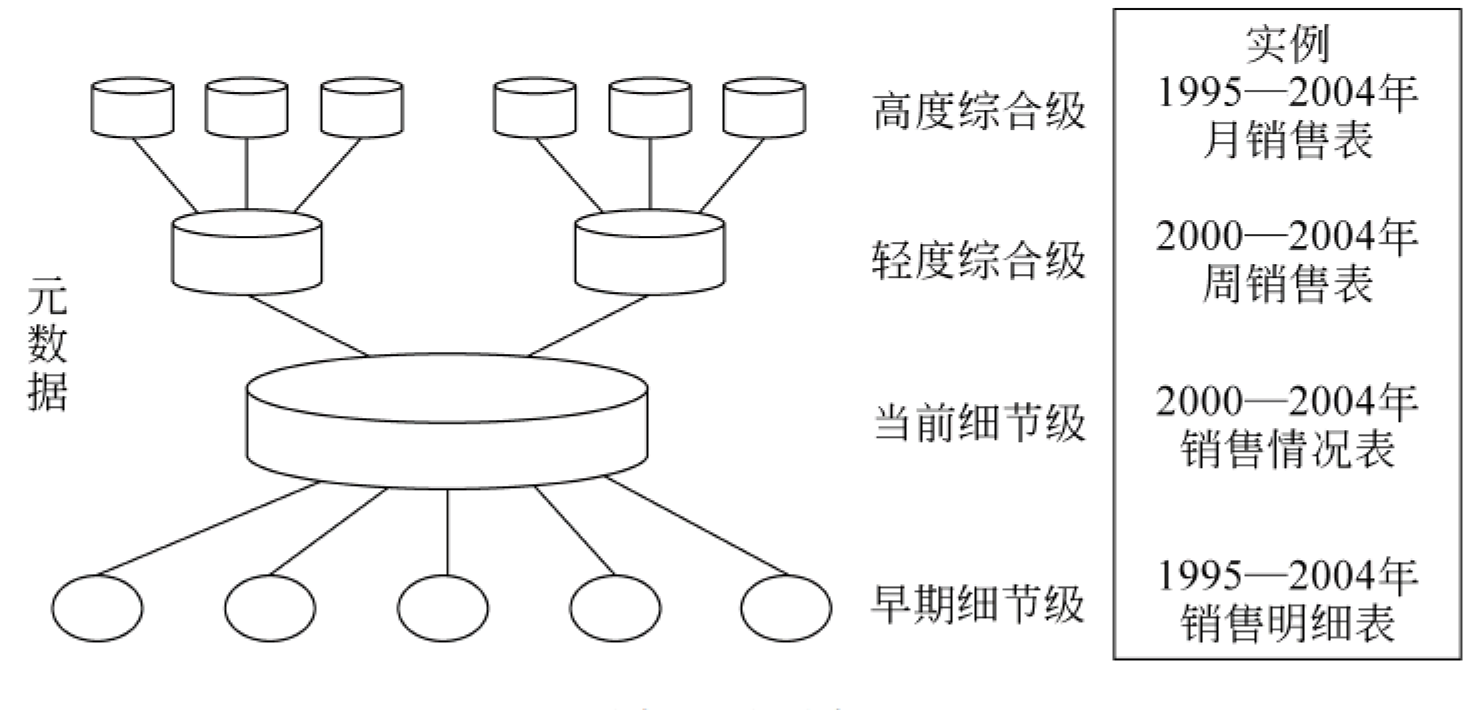


图 2-3 数据仓库数据组织结构

2.2.1 元数据

从图 2-3 可见,数据仓库组织结构中有一部分重要数据是元数据。元数据是“关于数据的数据”,如传统数据库中的数据字典就是一种元数据。在数据仓库中,元数据的内容比数据库中的数据字典更丰富、更复杂。元数据作为数据的数据,可对数据仓库中的各种数据进行详细的描述与说明,说明每个数据的上下文关系,使每个数据具有符合现实的真实含义,使最终用户了解这些数据之间的关系。

1. 元数据在数据仓库中的作用

(1) 为决策支持系统分析员和高层决策人员服务提供便利。数据仓库元数据的广义索引中存有每次数据装载时产生的有关决策的数据项,在做决策时,可以先查询该部分数据,再决定是否进行进一步的搜索。

(2) 解决面向应用的操作型环境和数据仓库的复杂关系。从面向应用的操作型环境到数据仓库的转换是复杂的多方面的,元数据包括对这种转换的描述。即包含了所有源数据项名、属性及其在数据仓库中的转换。

2. 元数据的使用

(1) 元数据在数据仓库开发期间的使用。数据仓库的开发过程是一个构造工程的过程,必须提供清晰的文档。这个过程产生的元数据主要描述 DW 目录表的每个运作的模式,数据的转化、净化、转移、概括和综合的规则与处理规则。

(2) 元数据在数据源抽取中的作用。元数据对多个来源的数据集成发挥着关键作用。利用元数据可以确定将数据源的哪些资源加载到 DW 中;跟踪历史数据结构变化过程;描述属性到属性的映射、属性转换等。

(3) 元数据在数据清理与综合中的使用。数据清理与综合负责净化资源中的数据、增加资源戳和时间戳,将数据转换为符合数据仓库的数据格式,计算和综合数据的值。元数据在这个过程中作为清理和综合数据的依据。

3. 元数据的分类

从不同的角度可以有多种分类。

(1) 按元数据的类型。按元数据的类型可以分为关于基本数据(数据源、数据仓库、应用程序管理)的元数据,用于数据处理(数据装载、更新处理、分析处理、数据抽取、转换、聚合规则等)的元数据和关于企业组织结构(用户、用户权限等)的元数据。

(2) 按抽象级别。按抽象级别可以分为概念级(业务的全部描述)、逻辑级(数据库的关系方案,逻辑多维模型等)和物理级(业务规则相应的 SQL 代码、关系的索引文件、分析应用的代码)的元数据。

(3) 按元数据承担的任务。按元数据承担的任务可以分为静态元数据(数据结构有关,如名称、格式等)和动态元数据(数据的状态与使用方法)。

(4) 从用户的角度。从用户的角度对元数据分类没有一个统一的标准,往往与元数据的使用目的相关。一般分为技术元数据和业务元数据两类。技术元数据是关于开发、维护和管理信息技术环境中所有的分析、设计、开发、管理等与技术关系密切的元数据;它是连接开发工具、应用程序和系统的技术纽带。业务元数据则使企业环境的服务更易于为终端用户所理解;它为业务目标和过程的解释提供便捷的浏览、导航和数据查询。

4. 元数据的内容

(1) 数据源的元数据如下:

① 每个来源的所有者描述信息;

② 每个来源的业务描述信息;

③ 原始来源的更新频率;

④ 每个来源使用的法律约束;

⑤ 存取方法、存取权利、特权,以及来源的存取口令;

⑥ 用来实现抽取过程的程序代码;

⑦ 自动抽取工具设置;

⑧ 特定抽取作业的结果信息,包括抽取时间、抽取内容以及完成情况。

(2) 数据模型的元数据如下:

① 企业概念模型;

② DW 数据模型;

③ 数据源到目标的映射。

(3) 数据准备区元数据如下:

① 数据传输调度以及特定传输的结果;

② 数据准备区文件使用情况;

③ 用于连接来源、删除字段、查找属性的作业规范;

④ 数据清洗规范;

⑤ 数据增强和映射转换;

⑥ DM 所要求的转换(比如解释空值的度量值);

⑦ 目标模式设计、来源到目标系统的数据流，目标数据的所有者；
⑧ 聚集定义、聚集使用统计、基本表使用统计；
⑨ 数据来源情况和审核检查记录(该记录真正来自何地、何时)；
⑩ 数据转换运行时间；
⑪ 数据转换软件的版本号；
⑫ 数据抽取处理的业务描述；
⑬ 有关抽取文件、软件以及元数据的安全性设置；
⑭ 数据传输的安全性设置；
⑮ 数据准备区的存档日志和恢复程序；
⑯ 数据准备区存档的安全性设置。

(4) 数据库管理系统元数据如下：
① 分区设置；
② 索引；
③ 数据库管理系统层次的安全性特权与授权；
④ 视图定义；
⑤ 存储过程与 SQL 管理脚本；
⑥ 数据库管理系统备份状态、备份程序以及备份安全性。

(5) 前台元数据如下：
① 业务名称和有关列、表以及分组的描述；
② 现有的查询和报告定义；
③ 连接规范；
④ 打印工具规范；
⑤ 最终用户文档；
⑥ 网络安全性用户特权概况；
⑦ 网络安全性身份验证证书；
⑧ 网络安全性使用统计，包括登录尝试、存取尝试以及按位置报告的用户标识符；
⑨ 个人用户概况；
⑩ 有关数据源、表、视图以及报告的使用及存取映射。

2.2.2　粒度的概念

粒度问题是数据仓库的一个重要概念，粒度是指数据仓库的数据单位中保存数据细化或综合程度的级别。粒度影响存放在数据仓库中的数据量大小，同时影响数据仓库所能回答查询问题的细节程度。粒度可以分为两种形式：按时间段综合数据的粒度和按采样率高低划分的样本数据库。

1. 按时间段综合数据的粒度

按时间段综合数据的粒度是对数据仓库中数据综合程度的高低度量，一般是按照不同的时间段来综合数据。它既影响数据仓库中数据量的多少，也影响数据仓库所能回答询问的种类。粒度越小，细节程度越高，综合程度越低，回答查询的种类就越多。反之，粒度的提高将会提高查询效率，但同时也造成回答细节问题能力下降。

为适应不同查询的需要，在数据仓库中经常是建立多重粒度，如图 2-3 的实例中有按周综合的轻度综合级数据和按月综合的高度综合级数据。

2. 样本数据库

与通常意义的粒度不同，样本数据库的粒度级别不是根据综合程度的不同来划分，而是根据采样率的高低来划分的。采样粒度不同的样本数据库可以具有相同的综合级别，一般它是以一定的采样率从细节档案数据或轻度综合数据中抽取的一个子集。

样本数据库不是一般目的的数据库，它是根据一定需求从源数据中的一个抽样。抽样的方法很多，一般是随机抽取。样本数据可以代替源数据进行模拟分析，经验证明，使用样本数据库可以大大降低实际分析的数据量，提高分析速度。并且，在保证一定抽样比例的情况下，如源数据量的 1/100 或 1/1000，得出的分析结果误差极小。分析的目的并不要求精确的结果，只需要建立起分析模型或是得到相对准确、能反映趋势的数据，从而验证用户的猜想，为下一步的策略确定方向或对当前分析程序作出相应调整，此时，样本数据库就大有用武之地。

样本数据库的抽取可以按照数据的重要程度不同来进行，样本数据库是建立在不同时点上的粒度。

2.2.3　分割问题

分割也是数据仓库中的一个重要概念，它是指将数据分散到各自的物理单元中，以便能独立处理，以提高数据处理效率。数据分割后的数据单元称为分片。分割之后，小单元内的数据相对独立，处理起来更快、更容易。

一般在进行实际的分析处理时，对于存在某种相关性的数据集合的分析是最常见的，如对某一时间或某一时段的数据的分析；对某一地区的数据的分析；对特定业务领域的数据的分析等；将具有这种相关性的数据组织在一起，就会提高效率。

数据分割的标准可以根据实际情况来确定，通常可选择按日期、地域或业务领域等来进行分割，也可以按多个分割标准的组合来进行，但一般情况分割标准应包括日期项。

1. 分割的优越性

分割之后有以下优点：

(1) 容易重构；

(2) 容易重组；

(3) 自由索引；

(4) 顺序扫描；

(5) 容易恢复；

(6) 容易监控。

数据仓库的本质之一就是灵活地访问数据，大块数据达不到这个目的。

2. 数据分割的标准

数据分割的标准是由开发人员选择的，在数据仓库中，按时间分割是必须进行的。具体有以下几种：

(1) 时间；

(2) 商业领域；

(3) 地理位置(区域);

(4) 组织单位(机构);

(5) 所有上述标准。

3. 分割的层次

分割的层次一般分为系统层和应用层两层。系统层的分割由数据库管理系统和操作系统完成;应用层的分割由应用程序完成,在应用层上分割更有意义。

2.2.4 数据仓库中的数据组织形式

数据仓库中的数据有多种组织形式,下面简单介绍几种数据仓库中常见的数据组织形式:简单堆积结构、轮转综合结构、简化直接结构和连续结构。

1. 简单堆积结构

简单堆积结构是数据仓库中最常用、最简单的数据组织形式。它从面向应用的数据库中每天的数据提取出来,然后按照相应的主题、集成为数据仓库中的记录。在此,关键是以"天"来进行集成并"堆积"。简单堆积结构数据组织形式如图 2-4 所示。

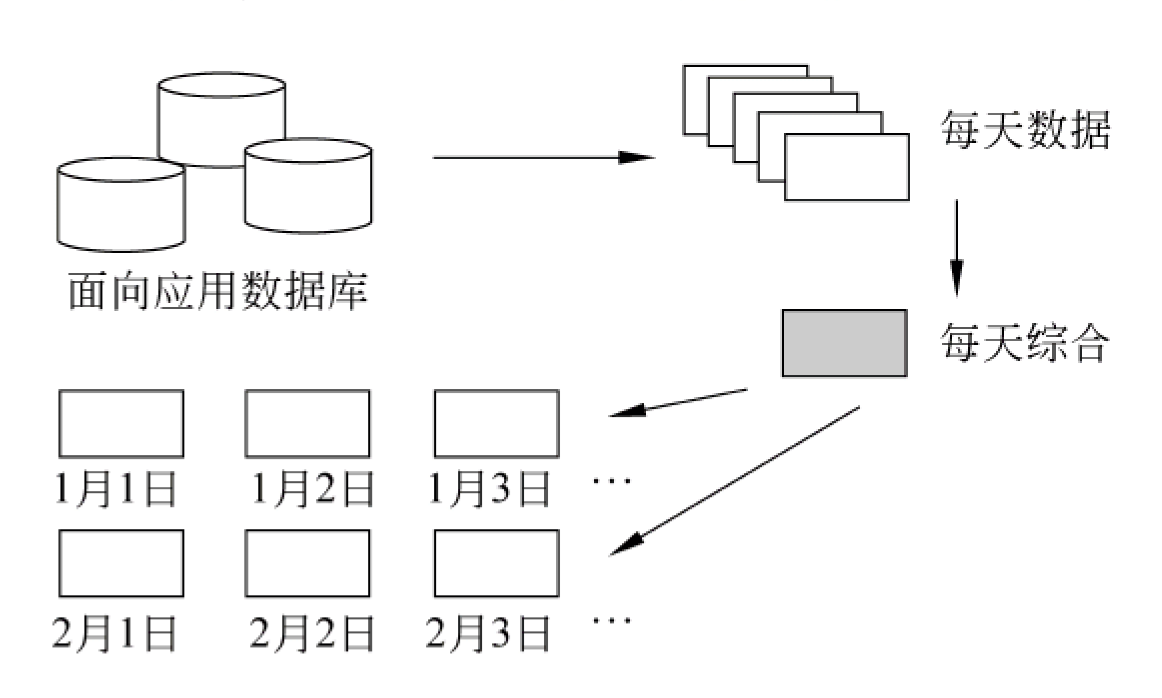


图 2-4 简单堆积结构数据组织形式

2. 轮转综合结构

在轮转综合结构中,数据存储单位被分为日、周、月、年等几个级别。在一星期的 7 天中,数据被逐一记录在每日数据集中;然后,7 天的数据被综合,记录在周数据集中;在下一个星期,日数据集被重新使用,以记录新数据。同理,周数据集达到 4 个后,数据再一次被综合并记入月数据集……以此类推。轮转综合结构简捷,数据量比简单堆积结构大大减少;但损失了数据细节,越早期的数据,细节损失越多。轮转综合结构如图 2-5 所示。

3. 简单直接结构

简单直接结构类似于简单堆积文件,但不是每天集成后放入数据仓库,而是间隔一定时间间隔,比如每隔一星期或一个月。简单直接结构也可以认为是按一定的时间间隔对数据库的采样。简单直接结构数据组织形式如图 2-6 所示。

4. 连续结构

通过两个或更多连续的简单直接结构数据组织形式的文件,可以生成另一种连续结构数据组织形式的文件。连续结构数据组织形式如图 2-7 所示。

对于各种文件结构的最终实现,在关系数据库中仍然要依靠"表"这种最基本的结构。

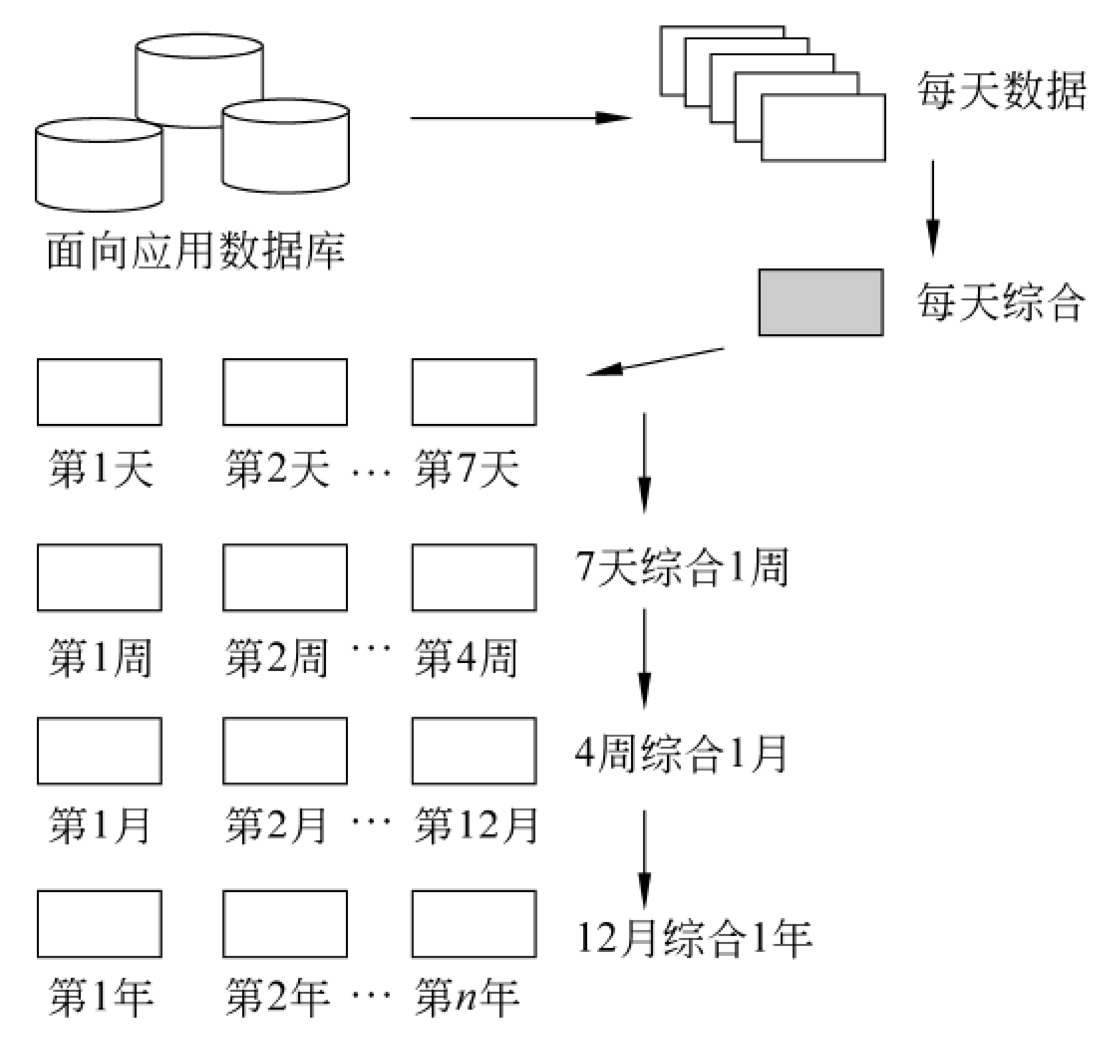


图 2-5 轮转综合结构数据组织形式

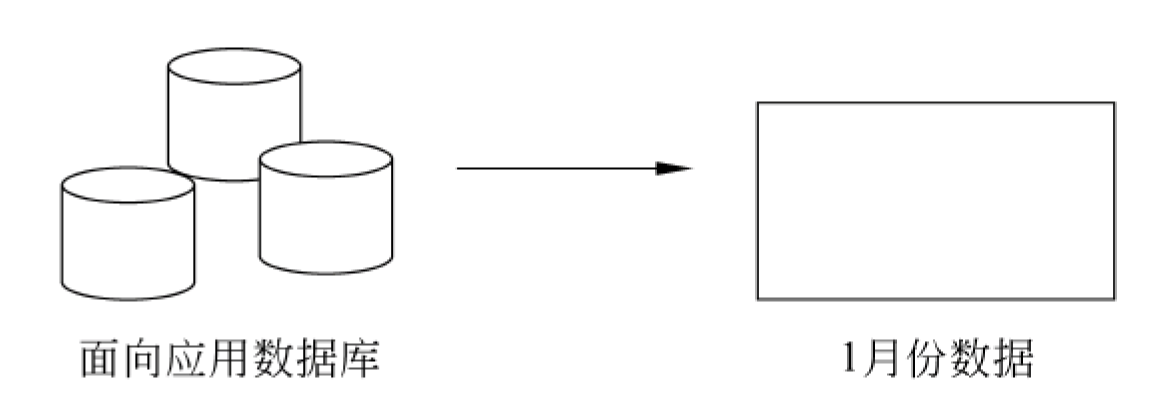


图 2-6 简单直接结构数据组织形式

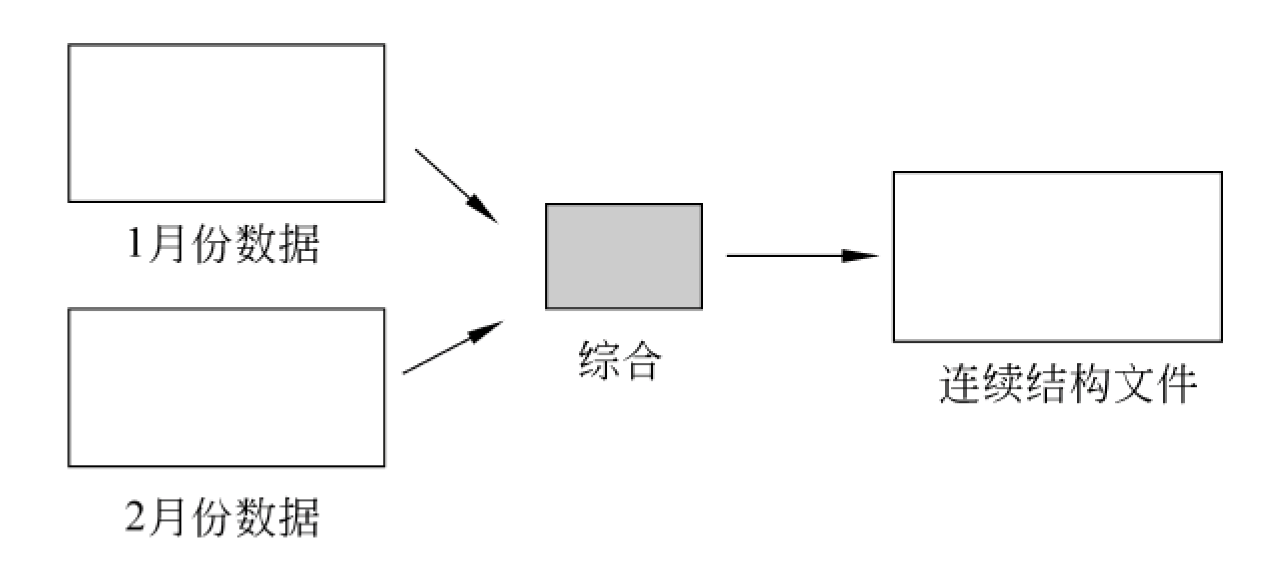


图 2-7 连续结构数据组织形式

2.3 数据仓库的数据模型

数据模型是对现实世界进行抽象的工具，抽象的程度不同，就形成不同抽象级别层次上的数据模型。数据仓库的数据模型与数据库的数据模型有所不同，主要表现如下：

(1) 数据仓库的数据模型中不包含纯操作型的数据；

(2) 数据仓库的数据模型扩充了码结构，增加了时间属性作为码的一部分；

(3) 数据仓库的数据模型中增加了一些导出数据。

上述 3 点差别也就是操作型环境中的数据与数据仓库中的数据之间的差别。虽然存在着这样的差别，在数据仓库设计中，同样存在着三级数据模型，即概念数据模型、逻辑数据模

型和物理数据模型。

2.3.1 概念数据模型

概念数据模型是主观与客观之间的桥梁,对计算机系统来说,概念数据模型是客观世界到机器世界的一个中间层次。人们首先将现实世界抽象为信息世界,然后将信息世界转化为机器世界,信息世界中的某一信息结构,就是概念数据模型。

概念数据模型最常用的表示方法是实体—联系(E-R)法,这种方法用 E-R 图作为它的描述工具,E-R 图描述的是实体以及实体之间的联系。由于 E-R 图具有良好的可操作性,形式简单,易于理解,便于与用户交流,对客观世界的描述能力较强,在数据库设计方面得到了广泛的应用。因为目前的数据仓库一般建立在关系数据库的基础上,为了和原有数据库的概念模型相一致,数据仓库的概念数据模型也采用 E-R 图描述。

E-R 图中的描述方法如下。

(1) 矩形。矩形表示实体,在数据仓库中表示主题,在矩形框内写上主题名。

(2) 椭圆形。椭圆形表示主题的属性,并用无向边把主题与其属性连接起来。

(3) 菱形。菱形表示主题之间的联系,菱形框内写上联系的名字。用无向边把菱形分别与有关的主题连接,在无向边旁标上联系的类型。若主题之间的联系也具有属性,则把属性和菱形也用无向边连接上。

如表 2-3 中,某商场的商品、顾客和供应商之间概念模型的 E-R 图可如图 2-8 所示。

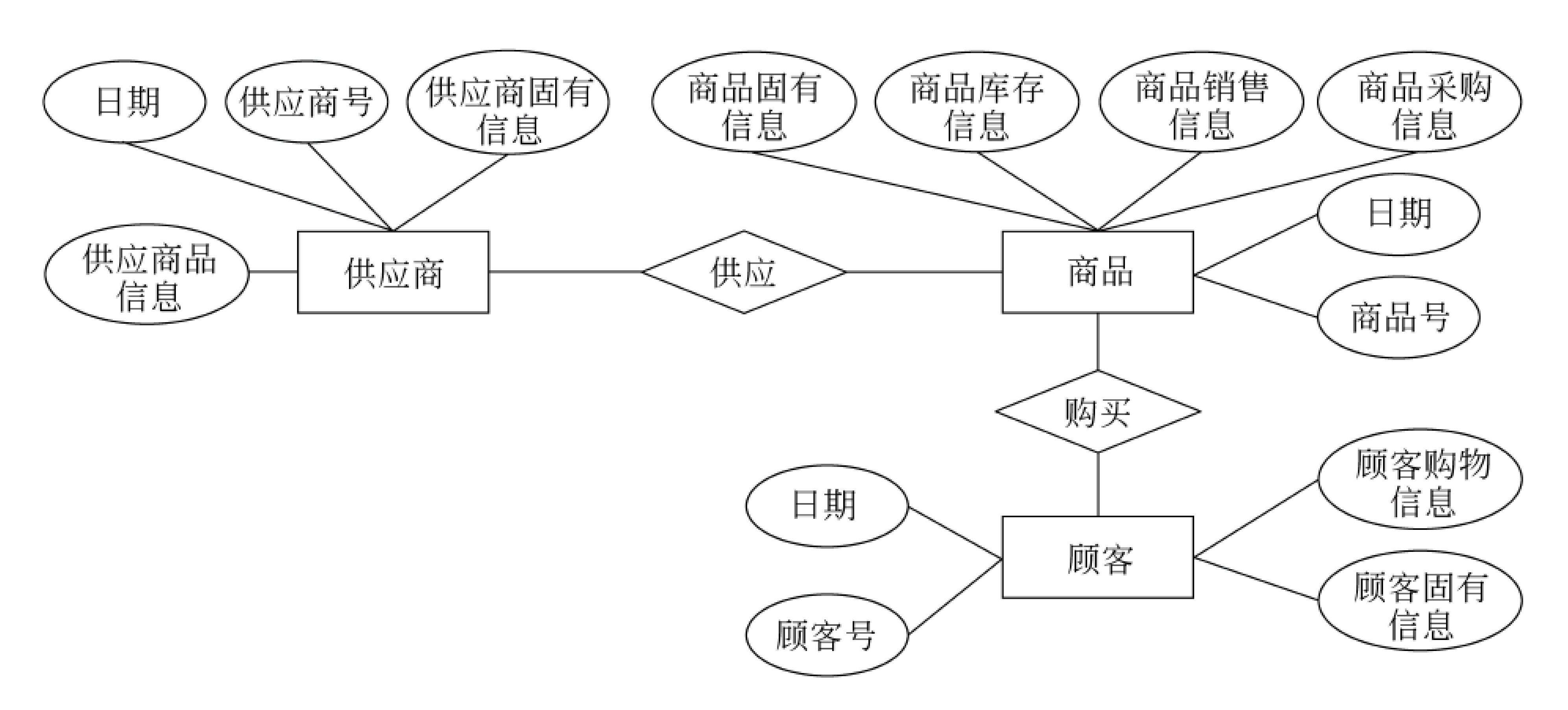


图 2-8 商品、顾客和供应商 E-R 图

2.3.2 逻辑数据模型

目前数据仓库一般建立在关系数据库基础之上,因此,在数据仓库的设计中采用的逻辑数据模型就是关系模型。无论是主题还是主题之间的联系都用关系来表示。

关系模型的主要概念如下。

(1) 关系:一个二维表。

(2) 元组:表中的一行称为一个元组。

(3) 属性:表中的一列称为属性,给每一列起一个名称即属性名。

(4) 主码:表中的某个属性组,其值唯一地标识一个元组。

(5) 域：属性的取值范围。

(6) 分量：元组中的一个属性组。

(7) 关系模式：对关系的描述，用关系名(属性名 1，属性名 2，…，属性名 n)表示。

数据仓库的逻辑数据模型描述了数据仓库主题的逻辑实现，即每个主题所对应关系表的关系模式的定义。

2.3.3 物理数据模型

数据仓库的物理数据模型就是逻辑数据模型在数据仓库中的实现，如物理存取方式、数据存储结构、数据存放位置以及存储分配等。物理数据模型是在逻辑数据模型的基础之上实现的，在进行物理数据模型设计实现时，所考虑的主要因素有：I/O 存取时间、空间利用率和维护代价。在进行数据仓库的物理数据模型设计时，考虑到数据仓库的数据量大但是操作单一的特点，可采取其他一些提高数据仓库性能的技术，如：合并表、建立数据序列、引入冗余、进一步细分数据、生成导出数据、建立广义索引等。

2.3.4 高层数据模型、中间层数据模型和低层数据模型

William H. Inmon 在《构建数据仓库》(Building the Data Warehouse)一书中提出了数据仓库三级数据模型的另一种提法：高层数据模型、中间层数据模型和低层数据模型。

1. 高层数据模型

高层数据模型对数据抽象程度最大，使用的主要表达工具是 E-R 图。首先确定 E-R 图所要集成的范围，并由各方用户提供自己的分 E-R 图，最后将各个分 E-R 图集成为整体的总 E-R 图。

2. 中间层数据模型

高层数据模型建好后，对高层数据模型中标识的每个主要的主题域或实体，都要建一个中间层数据模型。但中间层数据模型很难一次完全建好，需要不断扩展、完善。

中间层数据模型有以下 4 种基本构造。

(1) 连接数据组。连接数据组主要用于表示本主题域与其他主题域之间的联系，体现 E-R 图中实体之间的“关系”。一般情况下，连接数据组往往是一个主题的公共码键。

其他 3 种数据组主要按数据的稳定性来划分。

(2) 基本数据组。基本数据组的数据项是属于基本不会发生变化的项，如顾客号、顾客名、性别等有关顾客的固定描述信息的数据项。每个主题只存在一个基本数据组，基本数据组有属性和键码。主题的主码总是应包含在基本数据组中。

(3) 二次数据组。对于那些基本不变化，但又存在变化可能的数据项，归入二次数据组。如顾客的住址、文化程度、电话等数据项。二次数据组有对每个主要主题域可以存在多次的属性。

(4) 类型数据组。对于那些经常变化的数据项，归入类型数据组。如顾客的购物记录是变动频繁的数据项，所以归入类型数据组。

这种划分的好处是结构清晰，具有相似属性的数据被组织在一起；减少了冗余，如果将不变化或很少变化的数据项与经常变化的数据项混杂在一起存储，将产生大量冗余。

3. 低层数据模型

低层数据模型就是物理数据模型。

2.4　数据仓库设计步骤

数据仓库系统的原始需求不明确，并且在设计过程中会不断变化与增加，开发者最初并不能确切了解到用户明确而详细的需求，用户所能提供的无非是需求的大方向以及部分需求，不能较准确地预见到以后的需求。因为原型法的思想是从构建系统简单的基本框架着手，不断丰富与完善整个系统，因此，采用原型法来进行数据仓库的开发。但是，数据仓库的设计开发又不同于一般意义上的原型法，数据仓库的设计是数据驱动的。

数据仓库系统开发是一个经过不断循环、反馈而使系统不断增长与完善的过程，这也是原型法区别于系统生命周期法的主要特点。数据仓库的设计大体上可以分为以下几个步骤：

(1) 概念模型设计；

(2) 技术准备工作；

(3) 逻辑模型设计；

(4) 物理模型设计；

(5) 数据仓库生成；

(6) 数据仓库运行与维护。

这几个设计步骤的相互关系如图 2-9 所示。

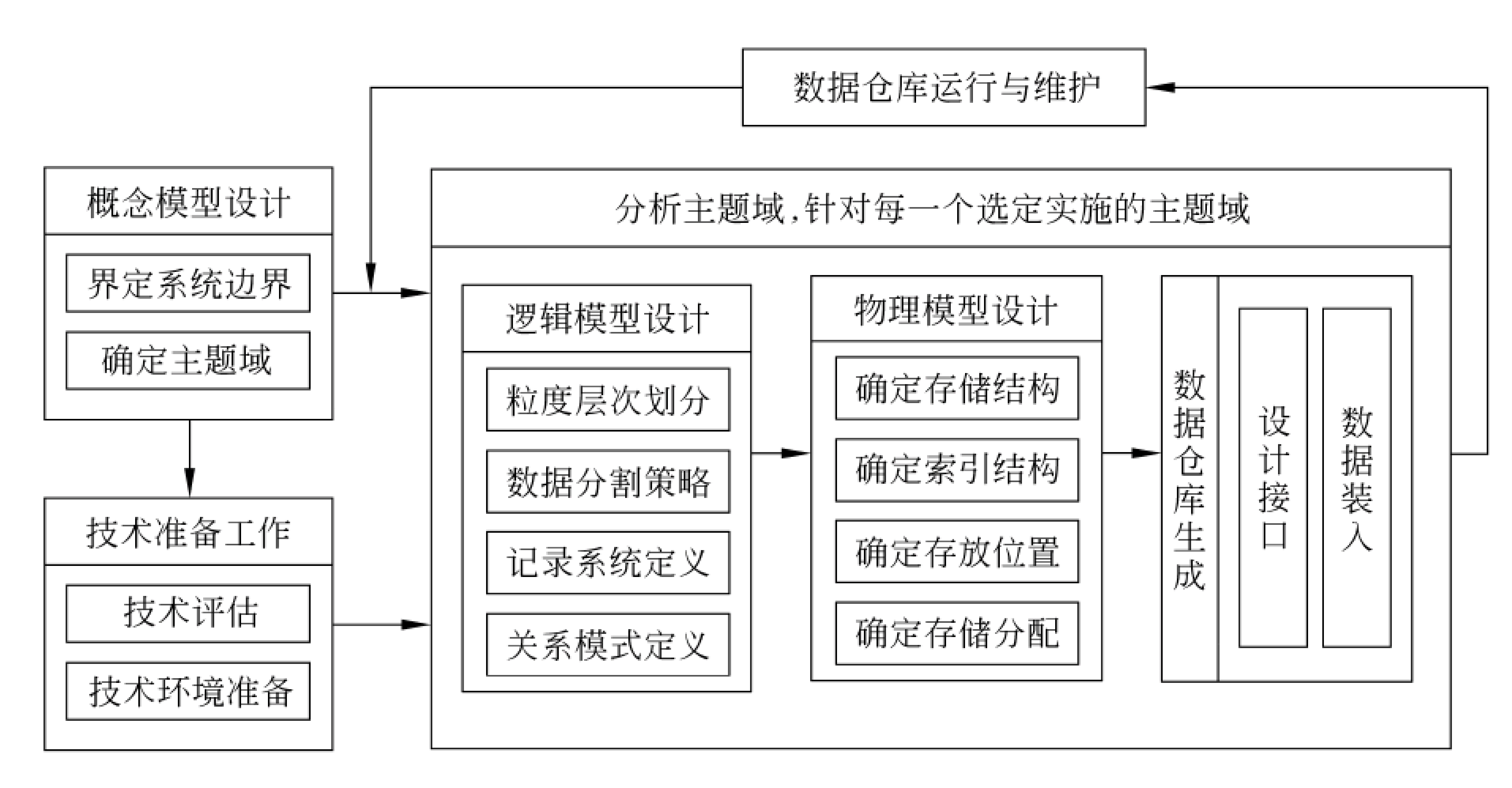


图 2-9　数据仓库设计步骤

下面就以图 2-9 所示的 6 个主要设计步骤为主线，介绍在各个设计步骤中设计的基本内容。

2.4.1　概念模型设计

进行概念模型设计所要完成的工作是界定系统边界和确定主要的主题域及其内容。概念模型设计的成果是在原有的数据库的基础上建立一个较为稳固的概念模型。

因为数据仓库是对原有数据库系统中的数据进行集成和重组而形成的数据集合，所以数据仓库的概念模型设计，首先要对原有数据库系统加以分析理解：原有的数据库系统中到底“有什么”、“怎样组织的”和“如何分布的”等，然后再来考虑应当如何建立数据仓库系统的概念模型。概念模型的设计是在较高的抽象层次上的设计，因此建立概念模型时不用考虑具体技术条件的限制。

1. 界定系统边界

界定系统边界前，首先了解下述需求：

(1) 要进行的决策类型有哪些；

(2) 决策者感兴趣的问题是什么；

(3) 这些问题需要什么信息；

(4) 要得到这些信息应该包含原有数据库系统的哪部分数据。

根据上述需求就可以界定一个大致的系统边界。因此，系统边界划分的前提是做好系统需求的调查。

2. 确定主要的主题域

根据需求调查和系统边界的界定，确定系统所包含的主题域，然后对每个主题域的内容进行较明确的描述。描述的内容包括：

(1) 主题域的公共键码；

(2) 主题域之间的联系；

(3) 代表主题的属性组。

3. 实例

以表 2-2 所示的某商场为例，分析概念模型设计的过程。

该商场已在各个部门建立了许多分散的数据库，分别处理各自的事物。如在人事、采购、库存、销售等几个部门分别存储着人事、采购、库存、销售的数据库。

首先，界定系统边界。了解到商场的领导最迫切的需求是准确地把握商场的经营状况，主要是商场的商品采购和销售情况。为此，需要分析：

(1) 顾客的购买趋势；

(2) 商品供应市场的变化趋势；

(3) 供应商信用等级。

根据以上分析的需求，得出所需数据包括：

(1) 商品销售数据；

(2) 商品采购数据；

(3) 顾客信息；

(4) 供应商信息。

据此，可以界定该系统的边界应该为原有的销售子系统、采购子系统和库存子系统。

在界定了系统边界的基础上，根据分析需求，确定该数据仓库系统包括 3 个主题域：商品、供应商和顾客。其中，顾客可购买多种商品(可能是不同供应商所供应)，一个供应商所供应的商品可被多个顾客购买；供应商提供多种商品，同一种商品也可由多个供应商供应。可以看出，这 3 个主题之间是由商品这一主题来相互联系，而顾客与供应商不发生直接的联系。也就是说，商品主题与供应商主题间的联系是商品供应关系，商品主题与顾客主题间的

联系是商品销售关系。

通过以上的界定系统边界和确定主要主题域后，初步画出该数据仓库系统的 E-R 图，如图 2-8。

2.4.2 技术准备工作

技术准备工作阶段的主要任务是进行技术评估和进行技术环境的准备。

进行技术评估，就是确定数据仓库的各项性能指标。一般情况下，需要确定的性能指标如下：

(1) 管理大数据量数据的能力；
(2) 进行灵活数据存取的能力；
(3) 根据数据模型重组数据的能力；
(4) 透明的数据发送和接收能力；
(5) 周期性成批装载数据的能力；
(6) 可设定完成时间的作业管理能力。

技术环境的储备主要是对系统软、硬件系统的准备。首先，根据以下数据确定软、硬件系统的配置：

(1) 预期在数据仓库上分析处理的数据量；
(2) 减少或减轻竞争性存取程序冲突的措施；
(3) 估算数据仓库的数据量；
(4) 估算进出数据仓库的数据通信量。

上述数据估算后，就可以进行软、硬件系统的配置。主要配置以下设备：

(1) 直接存取设备及管理软件；
(2) 网络；
(3) 操作系统；
(4) 对数据仓库进行查询、分析处理等操作的软件及界面；
(5) 管理数据仓库的软件。

所需配置的各种软件有些可以直接购买，有些需要请专业开发人员结合企业实际情况自主开发。

2.4.3 逻辑模型设计

逻辑模型设计的主要工作包括分析主题域，确定当前要装载的主题；确定粒度层次划分；确定数据分割策略；关系模式定义；记录系统定义。

1. 分析主题域

数据仓库的设计方法是一个逐步求精的过程，在进行设计时，一般是一次一个主题或一次若干个主题逐步完成，所以必须对概念模型设计步骤中确定的几个基本主题域进行分析，选择首先实施的主题域。选择第一个主题域所要考虑的是，它要足够大，以便使得该主题域能建设成为一个可应用的系统；它还要足够小，以便于开发和较快地实施。如果所选择的主题域很大并且很复杂，也可以先对一个有意义的子集进行开发。在每一次反馈过程中，都要进行主题域的分析。

例如，在前面所举的商场数据仓库设计的实例中，已经在设计概念模型中确定了3个主要主题域：商品、供应商和顾客。通过前面的分析，已经明确商品这一主题域与供应商和顾客主题域都有关系；通过对商品这一主题域，商场经营者就能对整个商场的经营状况有全面了解。因此，可以首先选定商品主题域来进行开发。

2. 划分粒度层次

数据仓库的粒度层次划分是数据仓库逻辑设计中要解决的一个重要问题。粒度层次划分是否适当直接影响到数据仓库中的数据量和所适合的查询类型。确定数据仓库的粒度层次的划分，可以使用第2.5.3小节中介绍的方法，通过估算数据行数和所需的存储空间数以及根据具体的分析需求来确定是采用单一粒度还是多重粒度，以及粒度划分的层次。如商场数据仓库的粒度层次可以划分为：早期细节级、当前细节级、轻度综合级和高度综合级等。

3. 确定数据分割策略

确定数据分割策略主要是选择适当的数据分割的标准。选择数据分割的标准一般要考虑数据量(不是记录行数)、数据分析处理的实际情况以及粒度划分策略等。数据量的大小决定是否进行数据分割和如何分割，数据分析处理的要求是选择数据分割标准的一个主要依据，还要考虑到所选择的数据分割标准是自然的、易于实施的；同时也要考虑数据分割的标准与粒度划分层次要相适应。

4. 定义关系模式

因为数据仓库的每个主题都是由多个表来实现的，这些表之间依靠主题的公共键码联系在一起。关系模式定义就是对在概念模式设计时确定的当前实施的主题域进行模式划分，以便形成多个表，并确定各个表的关系模式。如前面所列举商场的数据仓库设计例子中，已经确定商品这个主题是当前实施的主题域，对这个主题域进行关系模式定义如下。

(1) 公共键码：商品号。

(2) 商品固有信息：商品表。

(3) 商品采购信息：采购表1(细节级)，采购表2～n(按不同时间段的综合表)。

(4) 商品销售信息：销售表1(细节级)，销售表2～n(按不同时间段的综合表)。

(5) 商品库存信息：库存表1(细节级)，库存表2～n(按不同时间段的综合表)。

(6) 其他导出数据：其他数据表。

5. 定义记录系统

数据仓库中的数据来源于多个已经存在的操作型系统及外部系统。在将这些数据装载到数据仓库中时，必须选择最完整、最及时、最准确和最接近的数据作为记录系统，同时这些数据所在表的关系模式还应最接近构成主题的多个表的关系模式。另外，一定将记录系统的定义记入数据仓库的元数据中。

前面所举商场的实例中，“商品”主题的有关内容分散在原有的采购子系统、库存子系统、销售子系统的操作型部门数据库中。这三个数据源中有关商品的信息有相交的部分，可能存在不一致的信息，需要从记录系统的要求出发，选择原有的分散数据库中最完整、及时、准确和接近外部实体的数据定义为数据仓库的记录系统。在数据仓库的元数据中，应该记录主题名、属性名、数据源系统、源表名和源属性名等项。

2.4.4 物理模型设计

物理模型设计主要包括确定数据的存储结构、确定索引策略、确定数据存放位置、确定存储分配。

确定数据仓库实现的物理模型，要求设计人员必须深入了解以下内容：所选用的数据库管理系统，特别是存储结构和存取方法；了解数据环境，数据的使用频度、使用方式、数据规模以及响应时间要求；了解外部存储设备的分块原则，块大小的规定等特性以及设备的I/O 特性等。

1. 确定数据的存储结构

不同的存储结构有不同的实现方式，不同的适用范围和优缺点。设计人员在选择存储结构时应考虑存取时间、存储空间利用率和维护代价这 3 个方面的主要因素。

2. 确定索引策略

数据仓库中的数据量虽然很大，但其中的数据是不常更新的。因此，可以设计多种索引结构提高数据存取的效率，如广义索引。确定索引策略时，需要对数据的存取路径进行仔细地设计和选择。

3. 确定数据存放位置

在数据仓库系统中，同一个主题域的数据并不要求存放在相同的介质上。在物理设计时，要按数据的重要程度、使用频率以及对响应时间的要求进行分类，并将不同类的数据分配存储在不同的存储设备中。重要程度高、经常存取并对响应时间要求高的数据就存放在高速存储设备上，如硬盘；存取频率小或对存取响应时间要求低的数据可以放在低速存储设备上，如磁盘或磁带。

确定数据存放的位置时还应考虑以下因素：

(1) 是否进行合并表；

(2) 是否对一些经常性的应用建立数据序列；

(3) 对常用的、不常修改的表或属性是否冗余存储。

对于上述情况，应该记入数据仓库的元数据中。

4. 确定存储分配

存储分配主要包括块的大小、缓冲区大小和个数等，这些都应该在物理模型设计时确定；要根据数据库管理系统提供的参数和数据仓库所需要存放的数据量来决定。

2.4.5 数据仓库的生成

生成数据仓库主要是进行接口设计和将数据装入。数据装入后，还要在其上建立数据仓库的应用。

1. 接口设计

将操作型环境下的数据装载进入数据仓库环境，需要在两个不同环境的记录系统之间建立一个接口。接口应具有以下功能：

(1) 从面向应用和操作的环境生成完整的数据；

(2) 数据的转换；

(3) 数据的计算与综合；

(4) 对现有记录系统的有效扫描,以便以后进行追加。

还要考虑到物理设计的一些因素和技术条件限制,根据这些内容,制定规格说明,然后根据规格说明,进行接口编程。编程过程包括:伪码开发、编码、编译、检错和测试等步骤。

在接口编程时,应注意以下几方面:

(1) 保持高效性,这也是一般的编程所要求的;

(2) 要保存完整的文档记录;

(3) 要灵活,易于改动;

(4) 要能完整、准确地完成从操作型环境到数据仓库的数据抽取、转换与集成。

2. 数据装入

数据装入就是通过运行接口程序,将数据装入到数据仓库中去。在进行数据装入时,要完成以下工作:

(1) 确定数据装入的次序;

(2) 清除无效或错误数据;

(3) 数据粒度管理;

(4) 数据刷新等。

需要注意的是,在进行数据装入时,并不是一次就将准备装入的数据全部都装入数据仓库,而是按照逻辑模型设计中所确定和分析的主题域,先装入并生成某一主题域。这样可以尽快地进入下一步工作,在数据仓库的使用和维护中,尽早发现问题、提出新的需求,使设计不断完善、扩展。

2.4.6 数据仓库的使用和维护

数据仓库的使用和维护主要是开发决策支持系统 DSS 的应用;进一步理解需求,调整和完善数据仓库系统,维护数据仓库。

1. 开发 DSS 应用

在数据仓库环境中开发 DSS 应用与在操作型环境中开发 DSS 应用有着本质的区别。在数据仓库环境中的开发有以下几方面的特点:

(1) 数据仓库环境中开发 DSS 应用是从数据出发;

(2) 数据仓库环境中 DSS 应用的需求不能在开发初期完全了解;

(3) 数据仓库环境中 DSS 应用的开发是一个不断循环的过程,是启发式的开发。

DSS 应用分为例行分析处理和启发式分析处理两种类型。

(1) 例行分析处理。重复进行的分析处理,通常是属于部门级的应用,如部门统计分析、报表分析等。

(2) 启发式分析处理。企业经营者受到某种信息启发而进行的一些分析处理,随机性较大。

DSS 应用开发的大致步骤如图 2-10 所示。

各步骤简单说明如下。

第 1 步,确定所需的数据,从数据仓库中确定一个可能用到的数据范围。这是一个试探的过程。

第 2 步,编程抽取数据,根据第 1 步得到的数据范围,编写抽取程序获得这些数据。为

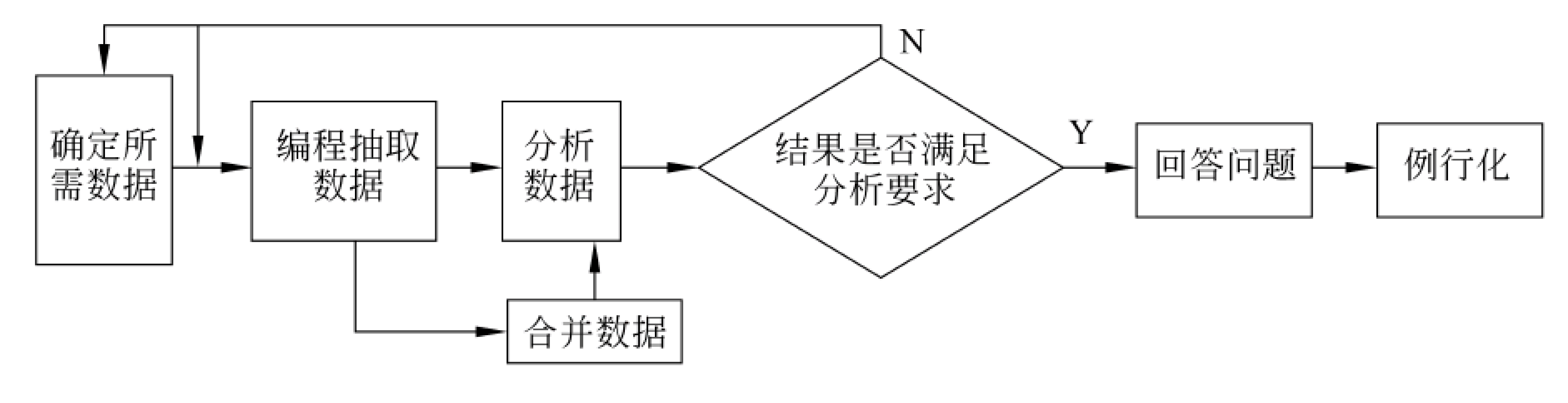


图 2-10 DSS应用开发步骤

适应分析需求多变的特点，要求所编写的抽取程序应通用、易于修改。

第 3 步，合并数据，当有多个数据抽取源时，将抽取来的数据进行合并、提炼，使数据符合分析处理的要求。

第 4 步，分析数据，对数据进行分析处理，并检查是否满足分析要求。如果不满足分析要求，返回到第 1 步，开始新一轮循环；若已经满足分析要求，继续进行下一步。

第 5 步，回答问题，在这一步要完成最终分析结果报告，一般情况需要进行多次循环得到。

第 6 步，例行化，最好对分析处理例行化，这样在以后进行同样的分析处理时，可以简化。例行化的另一个好处是不断积累例行处理，形成一个大的集合，可以进一步生成一个更大的系统。

2. 进一步理解需求，改善系统，维护数据仓库

数据仓库的开发使用逐步完善的原型法，原型法要求要尽快地让系统运行起来，尽早产生效益；要在系统运行或使用中，不断地理解需求，改善系统；不断地考虑新的需求，完善系统。

如前面所举商场建立数据仓库的过程，先以商品主题域为首先实施的主题域。在将商品主题的数据装入数据仓库后，就开发商品这一主题域的 DSS 应用，进行对商品主题的分析处理。在分析应用中，对原来的设计作出评价和调整。然后，就可以开发顾客、供应商等主题域的 DSS 应用。

维护数据仓库主要是管理日常数据的装入，包括刷新数据仓库的当前详细数据，将过时数据转化成历史数据，清除不再使用的数据，管理数据仓库的元数据等。

对于不同规模、不同应用的需求，以及不同的设计人员的开发习惯等，数据仓库的设计步骤并不是一成不变的，但最终应该满足用户的分析需求。

在 William H. Inmon 所著的《构建数据仓库》(Building the Data Warehouse)一书中介绍的数据仓库设计的步骤如图 2-11 所示。

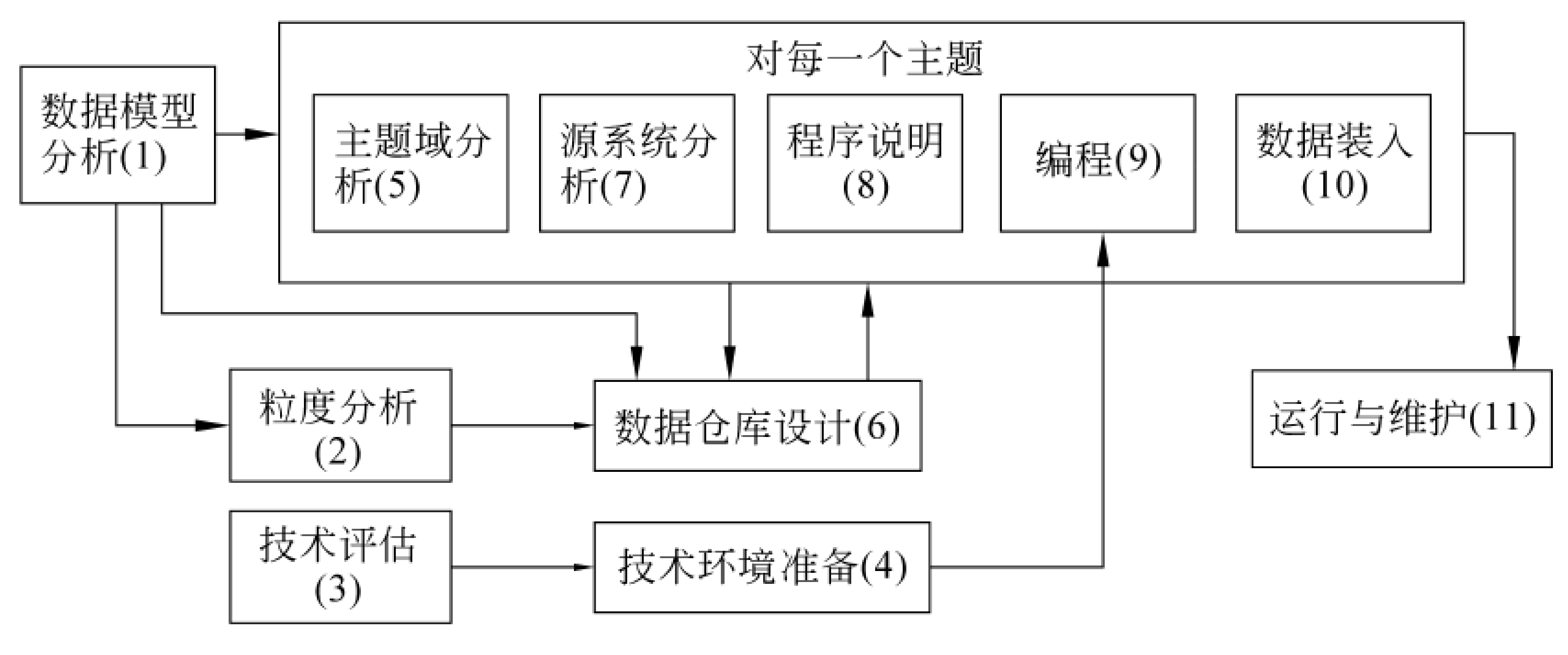


图 2-11 William H.Inmon 数据仓库设计步骤

2.5 利用 SQL Server 2005 构建数据仓库

2005 年底，微软公司正式推出 SQL Server 2000 的后继产品 SQL Server 2005。与前一代相比，SQL Server 2005 不仅提供了更加优秀的数据库管理功能，而且提供了一套完整的数据仓库和数据挖掘技术的解决方案。其中，SQL Server 2005 负责底层的数据库和数据仓库管理，SQL Server 2005 集成服务(SSIS)负责数据的抽取、转换和装载(ETL)，SQL Server 2005 分析服务负责 OLAP 分析和数据挖掘，SQL Server 2005 报表服务(SSRS)负责前端展示。

本节以订单分析为范例讲述利用 Microsoft SQL Server 2005 的分析服务来创建数据仓库，数据源使用现有的进销存数据库，其中主要使用到订单主表和订单从表，订单主表涉及的字段有订单编号、员工编号、采购商编号、订购日期、订单状态，订单从表涉及的字段有产品编号、产品名称、单价、数量、订单编号。订单编号作为订单主表的主键和从表的外键。

实现步骤如下。

(1) 打开 Visual Studio 2005 系统，使用菜单新建项目，如图 2-12 所示。



图 2-12 使用 Visual Studio 2005 系统新建项目

(2) 选择新建 Analysis Services 项目，将名称修改成“订单分析”，并且选择项目保存的位置，创建同名的解决方案，如图 2-13 所示。

(3) 打开解决方案资源管理器，查看已经创建的解决方案，使用鼠标在“数据源”处右击，选择“新建数据源”，并且在数据源向导界面单击“下一步”按钮，如图 2-14 和图 2-15 所示。

(4) 选择如何连接数据源，选择“基于现有连接或新连接创建数据源”，单击“新建”按钮，如图 2-16 所示。

(5) 在打开的连接管理器窗口选择数据库，提供程序选择“本机 OLE DB\SQL Native Client”，服务器名为已经连接的服务器名，选择登录到服务器的身份验证方式(与数据库服



图 2-13　新建 Analysis Services 项目

图 2-14　新建数据源



图 2-15　新建数据源向导



图 2-16　选择如何连接数据源

务器的设置相关),并选择数据库名,然后单击“确定”按钮,如图 2-17 和图 2-18 所示。



图 2-17　连接管理器

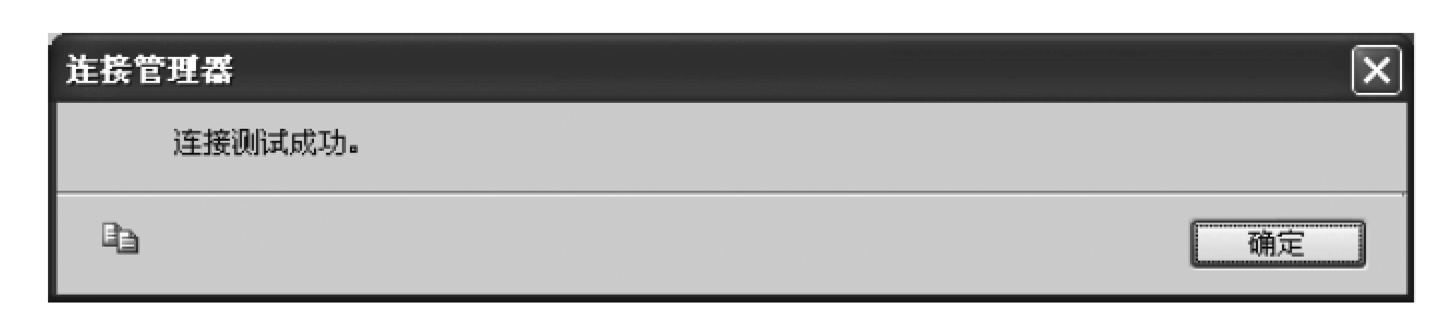


图 2-18　连接管理器连接测试成功窗口

(6) 单击图 2-18 窗口的“确定”按钮后回到“选择如何定义连接”窗口,选中已连接的数据库并单击“下一步”按钮,如图 2-19 所示。



图 2-19　选择已经连接的数据库作为数据源

(7) 选择分析服务器使用“使用服务账户”作为连接数据源的凭证，并单击“下一步”按钮，如图 2-20 所示。



图 2-20　选择连接数据源的凭证

(8) 完成新建数据源向导并确定数据源的名称，如图 2-21 所示。



图 2-21　新建数据源向导完成

(9) 使用解决方案管理器新建数据源视图，打开向导并单击“下一步”按钮，如图 2-22 和图 2-23 所示。

(10) 选择已经创建的数据源作为视图的数据源，如图 2-24 所示。

(11) 选择数据源中的表和视图，如图 2-25 所示。

(12) 完成数据源视图向导并为该视图命名，如图 2-26 所示。

(13) 新建多维数据集，打开多维数据集向导并单击“下一步”按钮，如图 2-27 和图 2-28 所示。

(14) 选择多维数据集的生成方法，本范例选择“使用数据源生成多维数据集”，并选中“自动生成”和“创建属性和层次结构”，如图 2-29 所示。

图 2-22　右击新建数据源视图



图 2-23　新建数据源视图向导



图 2-24　选择视图的数据源



图 2-25　选择表和视图



图 2-26　完成新建数据源视图向导



图 2-27　新建多维数据集



图 2-28　多维数据集向导



图 2-29　选择生成多维数据集的方法

(15) 选择多维数据集的数据源视图并单击“下一步”按钮,如图 2-30 所示。



图 2-30　选择多维数据集的数据源视图

(16) 单击“下一步”按钮,检测事实数据表和维度表,如图 2-31 所示。



图 2-31　检测事实数据表和维度表

(17) 单击“下一步”按钮,标识数据源视图中的事实表和维度表,如图 2-32 所示。

(18) 选择多维数据集的度量值,如图 2-33 所示。

(19) 单击“下一步”按钮,扫描维度,检测层次结构,如图 2-34 所示。



图 2-32　标示事实表和维度表



图 2-33　选择度量值



图 2-34　扫描维度

(20) 单击“下一步”按钮,查看维度结构,并可以做适当修改,如图 2-35 所示。



图 2-35　查看维度结构

(21) 单击“下一步”按钮,完成多维数据集向导,如图 2-36 所示。



图 2-36　完成多维数据集向导

(22) 创建完数据仓库,如图 2-37 所示,最后可以生成该数据仓库并部署,如图 2-37 所示。

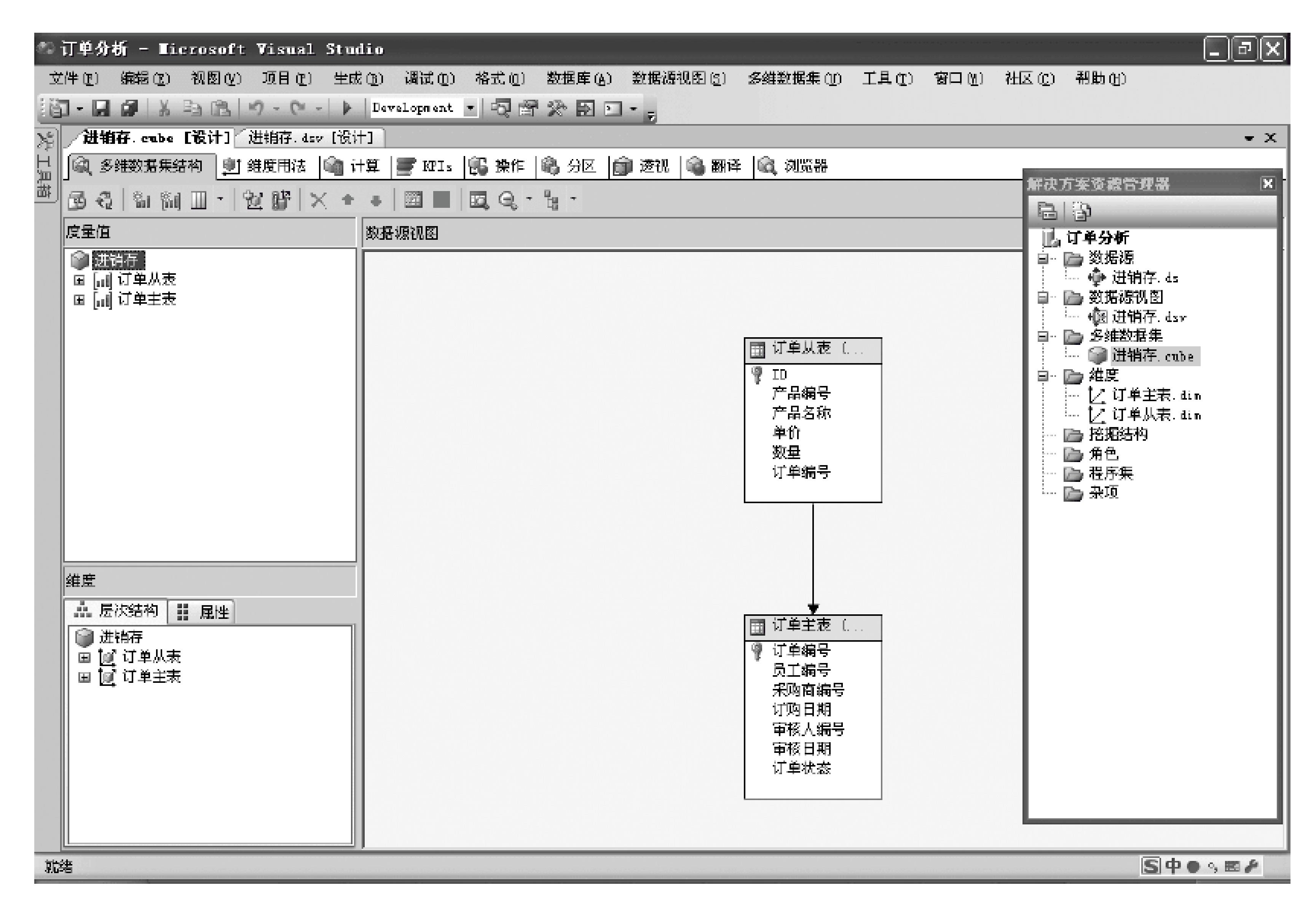


图 2-37 创建完成数据仓库界面

小结

本章首先对数据仓库的概念作了较深入的讲解。对数据仓库的数据是面向主题的、是集成的、是不可更新的以及是随时间不断变化的定义,逐条进行了讲述。然后对数据仓库的结构中的元数据、粒度问题及分割问题进行了讲解,并描述了数据仓库中的数据组织形式。接下来对数据仓库系统的设计从设计方法、数据模型以及提高数据仓库的性能方面进行了较详细的介绍,并给出了数据仓库设计步骤。本章的最后讲述了利用 SQL Server 2005 构建数据仓库。读者通过本章的学习应该对数据仓库的定义有较深入的理解,重点掌握数据仓库的粒度概念,掌握如何提高数据仓库的性能以及数据仓库设计方法与设计步骤,学会使用 SQL Server 2005 构建数据仓库。

习题 2

1. 如何理解数据仓库是面向主题的、集成的、不可更改的和是随时间不断变化的。

2. 什么叫元数据? 起什么作用?

3. 如何理解数据仓库中的粒度的概念? 如何确定数据仓库的粒度? 数据量与粒度有什么关系?

4. 为什么要进行数据仓库的清理? 如何清理?

5. 数据仓库设计有哪 3 级数据模型? 各如何设计?

6. 采用什么方法可以提高数据仓库的性能?

7. 叙述数据仓库设计的具体步骤,如何实现?

8. 什么是数据仓库的直接访问,什么是数据仓库的间接访问? 两者有何不同?

9. 数据仓库有哪些应用领域? 各举例说明。

第 3 章　联机分析处理技术

随着计算机技术的广泛应用，企业每天都要产生大量的数据，如何从这些数据中提取对企业决策分析有用的信息，是企业决策管理人员所面临的一个难题。传统的数据库系统即联机事务处理系统(online transaction processing，OLTP)，作为数据管理手段，主要用于事务处理，但它对分析处理的支持一直不能令人满意。因此，人们逐渐尝试对 OLTP 数据库中的数据进行再加工，形成一个综合的、面向分析的环境，以更好地支持决策分析。数据仓库和联机分析处理(online analysis processing，OLAP)是决策支持系统的有机组成部分。数据仓库从分布在企业内部各处的 OLTP 数据库中提取数据并对所提取的数据进行预处理，为企业决策分析提供所需的数据；OLAP 则利用存储在数据仓库中的数据完成各种分析操作，并以直观易懂的形式将分析结果返回给决策分析人员。

3.1　OLAP 概述

3.1.1　OLAP 的由来

在过去的二十几年中，大量的企业利用关系型数据库来存储和管理业务数据，并建立相应的应用系统来支持日常业务运作。这种应用以支持业务处理为主要目的，被称为联机事务处理，它所存储的数据被称为操作型数据或业务数据。

随着数据库技术的广泛应用和市场竞争的日趋激烈，企业更加强调决策的及时性和准确性。传统的联机事务处理系统作为数据管理的手段，对于分析处理的支持不能满足决策管理者对数据库进行复杂分析和获取直观易懂的查询结果的要求，因此，以支持决策管理分析为主要目的的应用迅速崛起。人们开始尝试对 OLTP 数据库中的数据进行再加工，形成一个综合的、面向分析的、更好的支持决策制定的决策支持系统(decision support system，DSS)。因此，Codd 提出了多维数据库和多维分析的概念，即联机分析处理。

3.1.2　OLAP 的一些基本概念

(1) 维(dimension)。维是人们观察数据的特定角度。例如，企业常常关心产品销售随时间的变化情况，这是从时间的角度来观察产品的销售，因此时间就是一个维。又例如银行会给不同经济性质的企业贷款，如国有企业、集体企业等，若从企业性质的角度来分析贷款数据，那么经济性质也就成了一个维度。

(2) 维层次(level)。人们观察数据的某个特定角度(即某个维)还可以存在细节程度不同的各个描述方面(时间维：日期、月份、季度、年)，称这多个描述方面为维的层次。

(3) 维成员(member)。维的一个取值称为该维的一个维成员，是数据项在某维中位置的描述(如“某年某月某日”是在时间维上位置的描述)。如果一个维是多层次的，那么该维的维成员是在不同维层次的取值组合。

(4) 多维数据集。多维数据集是决策支持的支柱，也是OLAP的核心，有时也称为立方体或超立方体。三维数据可以利用三维坐标建立立方体进行表示，超三维数据可以利用一个多维表来进行显示。

(5) 数据单元。在多维数据集中每个维都选定一个维成员以后，这些维成员的组合就唯一确定了一个数据单元(维1维成员，维2维成员，维3维成员，…)。

(6) 多维数据集的度量值：在多维数据集中有一组度量值，这些值是基于多维数据集中事实表的一列或多列数字。度量值是多维数据集的核心值，是最终用户在数据仓库应用中所需要查看的数据。

3.1.3　OLAP的定义与特征

OLAP委员会对联机分析处理的定义为，使分析、管理或执行人员能够从多种角度对从原始数据中转化出来的、能够真正为用户所理解的、并真实反映企业维特性的信息进行快速、一致、交互地存取，从而获得对数据更深入了解的一类软件技术。

联机分析处理的用户是企业中的专业分析人员及管理决策人员，他们在分析业务经营数据时，从不同的角度来审视业务的衡量指标是一种很自然的思考模式。例如分析销售数据，可能会综合时间周期、产品类别、分销渠道、地理分布、客户群类等多种因素来考量。而联机分析处理就是直接仿照用户的多角度思考模式，预先为用户组建多维的数据模型。这里，维是指用户的分析角度。一旦多维数据模型建立完成，用户可以快速地从各个分析角度获取数据，也能动态地在各个角度之间切换或者进行多角度综合分析，从而具有极大的分析灵活性。这也是联机分析处理在近年来被广泛关注的根本原因，它从设计理念和真正实现上都与旧有的管理信息系统有着本质的区别。其主要特征概括如下：

(1) 快速性。用户对OLAP的快速反应能力有很高的要求，系统应能在5秒内对用户的大部分分析要求做出反应，这也是OLAP的一个显著特点。

(2) 可分析性。OLAP系统应能处理与应用有关的任何逻辑分析和统计分析，用户无须编程就可以定义新的计算，将其作为分析的一部分，并以用户理想的方式给出报告。用户可以在OLAP平台上进行数据分析，也可以连接到其他外部分析工具上，如时间序列分析工具、成本分配工具、意外报警、数据开采等。

(3) 多维性。它是OLAP的关键属性。系统必须提供对数据分析的多维视图和分析，包括对层次维和多重层次维的完全支持。事实上，多维分析是分析企业数据最有效的方法，是OLAP的灵魂。

(4) 信息性。不论数据量有多大，也不管数据存储在何处，OLAP系统应能及时获得信息，并且能管理大容量信息。这里有许多因素需要考虑，如数据的可复制性、可利用的磁盘空间、OLAP产品的性能及与数据仓库的结合度等。

3.2　OLAP中的多维分析操作

OLAP的基本多维分析操作有钻取(drill-up和drill-down)、切片(slice)和切块(dice)以及旋转(pivot)等。

3.2.1　钻取

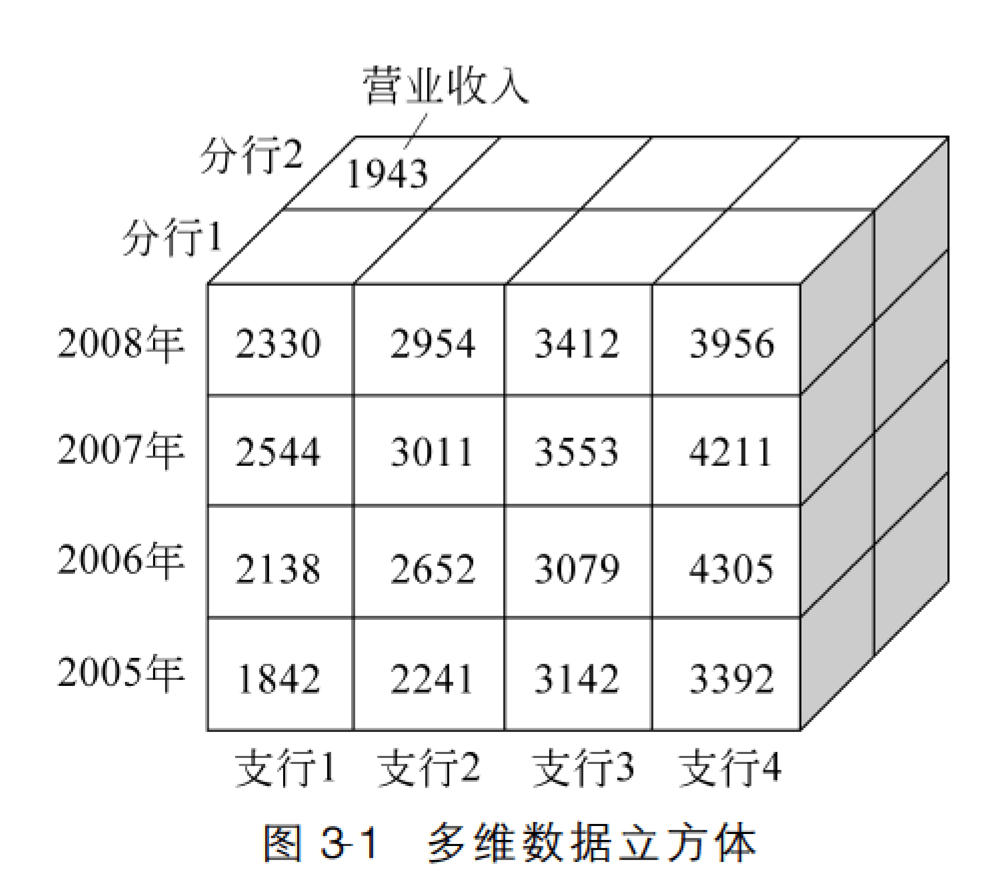


图 3-1　多维数据立方体

钻取是改变维的层次,变换分析的粒度。它包括向下钻取(drill-down)和向上钻取(drill-up)。drill-up是在某一维上将低层次的细节数据概括到高层次的汇总数据,或者减少维数;而Drill-down则相反,它从汇总数据深入到细节数据进行观察或增加新维。例如:图3-1所示的数据立方体经过沿着银行分行维的概念层次上卷,由银行分行上升到城市,得到图3-2所示的立方体;图3-1所示的数据立方体经过沿时间维下钻,由年度下降到2008年的各个季度,得到图3-3所示的数据立方体。

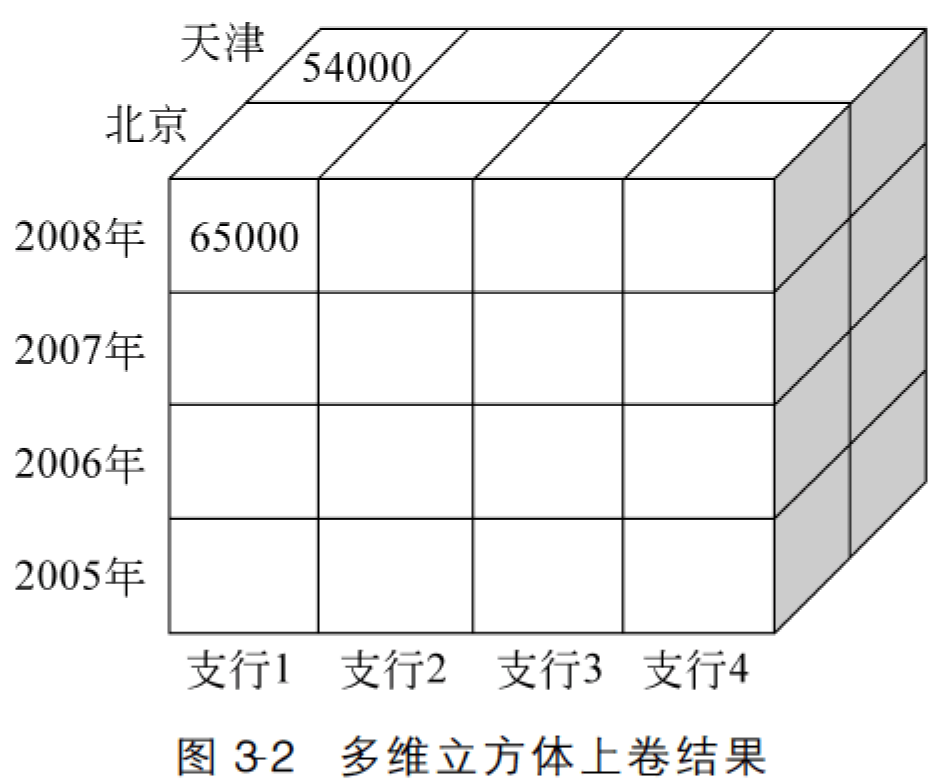


图 3-2　多维立方体上卷结果

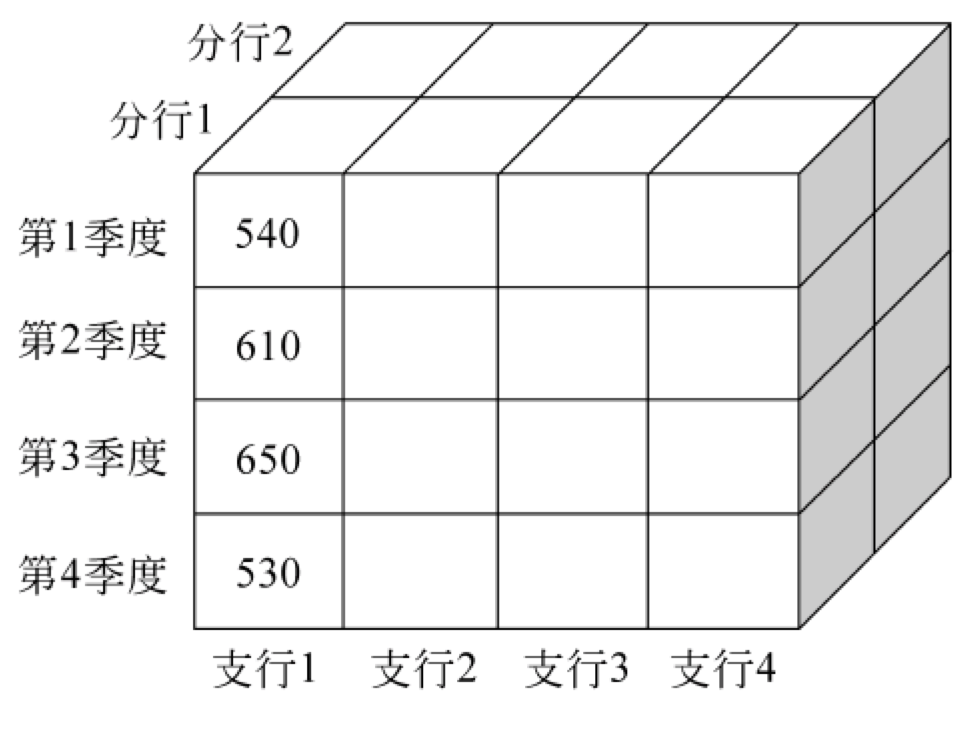


图 3-3　多维立方体下钻后结果

3.2.2　切片和切块

(1) 切片。在给定数据立方体的一个维上进行选择操作就是切片,切片的结果是得到一个二维平面数据。例如,对图3-1所示数据立方体,使用条件:

```
银行分行="分行 1"
```

进行选择,就相当于在原来的立方体中切出一片,结果如图3-4所示。

(2) 切块。在给定数据立方体的两个或多个维上进行选择操作就是切块,切块的结果得到一个子立方体。例如,对图3-1所示数据立方体,使用条件:

```
(银行分行="分行 1"OR"分行 2")
AND (时间="2007 年"OR"2008 年")
AND(银行支行="支行 1"OR"支行 2")
```

进行选择,就相当于在原立方体中切出一小块,结果如图3-5所示。

3.2.3　旋转

旋转是变换维的方向,即在表格中重新安排维的放置(例如行列互换)。例如,表3-1给



图 3-4　多维立方体切片结果

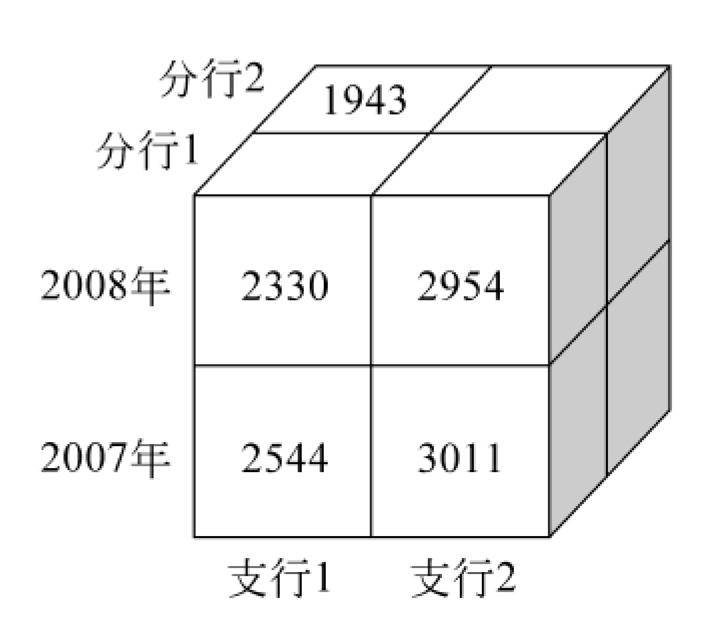


图 3-5　多维立方体切块结果

出的是按部门、年、季度排列的结果，而表 3-2 是按年、季度、部门排列的结果。

表 3-1　部门、年、季度排列表

	2008 年				2009 年			
	一季度	二季度	三季度	四季度	一季度	二季度	三季度	四季度
部门一	50	60	70	80	40	50	60	70
部门二	60	65	75	85	45	55	65	75
部门三	90	100	105	120	80	75	55	60

表 3-2　部门、季度、年排列表

		部门一	部门二	部门三
2008 年	一季度	50	60	90
	二季度	60	65	100
	三季度	70	75	105
	四季度	80	75	120
2009 年	一季度	40	45	80
	二季度	50	55	75
	三季度	60	65	55
	四季度	70	75	60

图 3-6 是图 3-1 所示立方体通过旋转横纵坐标得到的立方体。

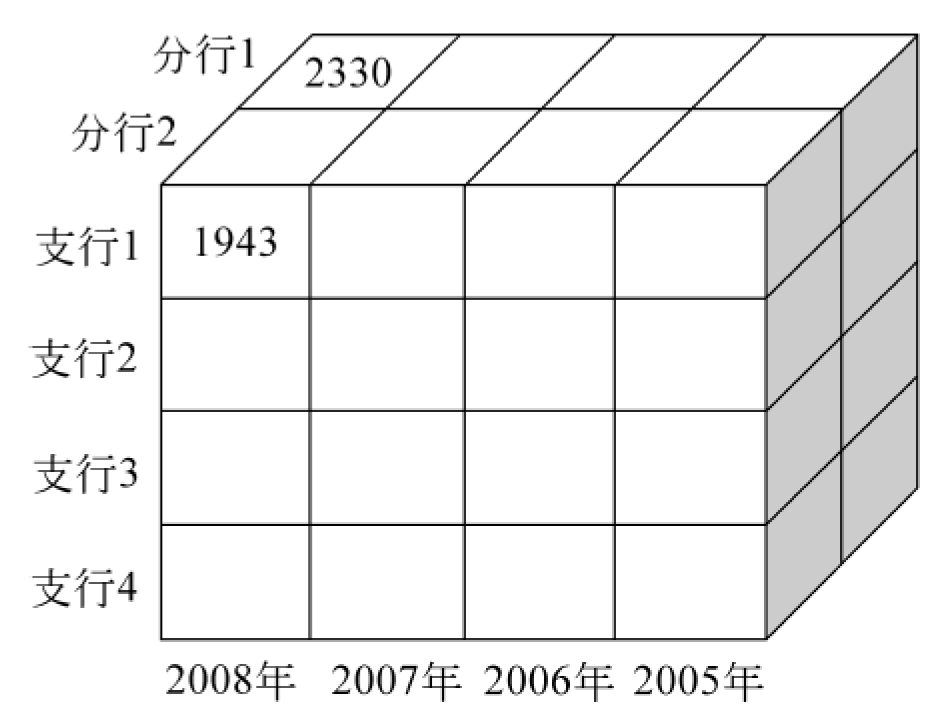


图 3-6　多维立方体横纵坐标转轴结果

3.3 OLAP 的基本数据模型

OLAP 系统一般以数据仓库作为基础，从数据仓库中抽取详细数据的一个子集，经过必要的聚集存储到 OLAP 存储器中供前端分析工具读取。为了保证信息处理所需的数据以合适的粒度、合理的抽象程度和标准化程度存储，按照其数据存储格式可以分为关系 OLAP(relational OLAP，ROLAP)、多维 OLAP(multidimensional OLAP，MOLAP)和混合型 OLAP(hybrid OLAP，HOLAP)这 3 种类型。

3.3.1 多维联机分析处理

MOLAP 利用一种专有的多维数据库来存储 OLAP 分析所需要的数据，数据采用 n 维数组的多维方式存储，形成“立方体”的结构，并以多维视图的方式显示。MOLAP 存储模式将数据与计算结果都存储在立方体结构中，即将多维数据集区的聚合、维度、汇总数据以及其源数据的副本等信息均以多维结构存储在分析服务器上。

1. MOLAP 的创建步骤

确定分析功能：在筹建 MOLAP 的时候首先要选择分析的功能是什么。

确定分析值：根据功能的选择，确定相应的分析数值。

构造分析维：构造分析维，即确定从哪些角度来分析这些分析数值。

定义逻辑模型：在确定了 MOLAP 的分析对象、分析角度及详略程度之后就可以定义逻辑模型和多维数据存储的方式。

2. MOLAP 的功能

(1) 与多维数据库进行交互的功能。可以与数据库中的信息进行交互，从而在分析决策中完成预测、预算、计划。

(2) 快速反应的功能。MOLAP 的快速反应功能可以为用户提供良好的联机分析环境。

(3) 挖掘信息间内在联系的功能。MOLAP 利用强大的计算引擎和比较分析，分析数据库中各种信息之间的微妙关系。

3. MOLAP 的优缺点

MOLAP 结构的主要优点是它能迅速地响应决策分析人员的分析请求并快速地将分析结果返回给用户，其缺点是限制了 MOLAP 结构的灵活性，主要表现在以下几方面。

(1) 用户很难对维数进行动态变化。每增加一维都会使多维数据库的规模急剧增加，所需的预处理时间也会大大增加。

(2) 对数据变化的适应能力较差。当数据或计算频繁变化时，其重复计算量相当大，有时还需重新构建多维数据库。

(3) 处理大量细节数据的能力差。由于 MOLAP 的预处理能力较强，这就限制了它处理大量细节数据的能力。

3.3.2 关系联机分析处理

1. ROLAP 的数据模型

ROLAP 的底层数据库是关系型数据库，其数据以及计算结果均直接由关系数据库获得，并且以关系型的结果进行多维数据的表示和存储。在 ROLAP 中，数据的预处理程度一般不高，但是灵活性高；用户可以动态定义统计和计算方式，可移植性好。ROLAP 一般采用星状模式（star schema）或雪花状模式（snowflake schema）来表达多维数据视图。

（1）星状模式。这是一种最常见的模型范例，其包括一个大的包含大批数据并且不含冗余的中心表（事实表）；一组小的维表，每维一个。这种模式图很像星光四射，维表围绕中心事实表显示在射线上。例如，图 3-7 表示的是一个星型模式。

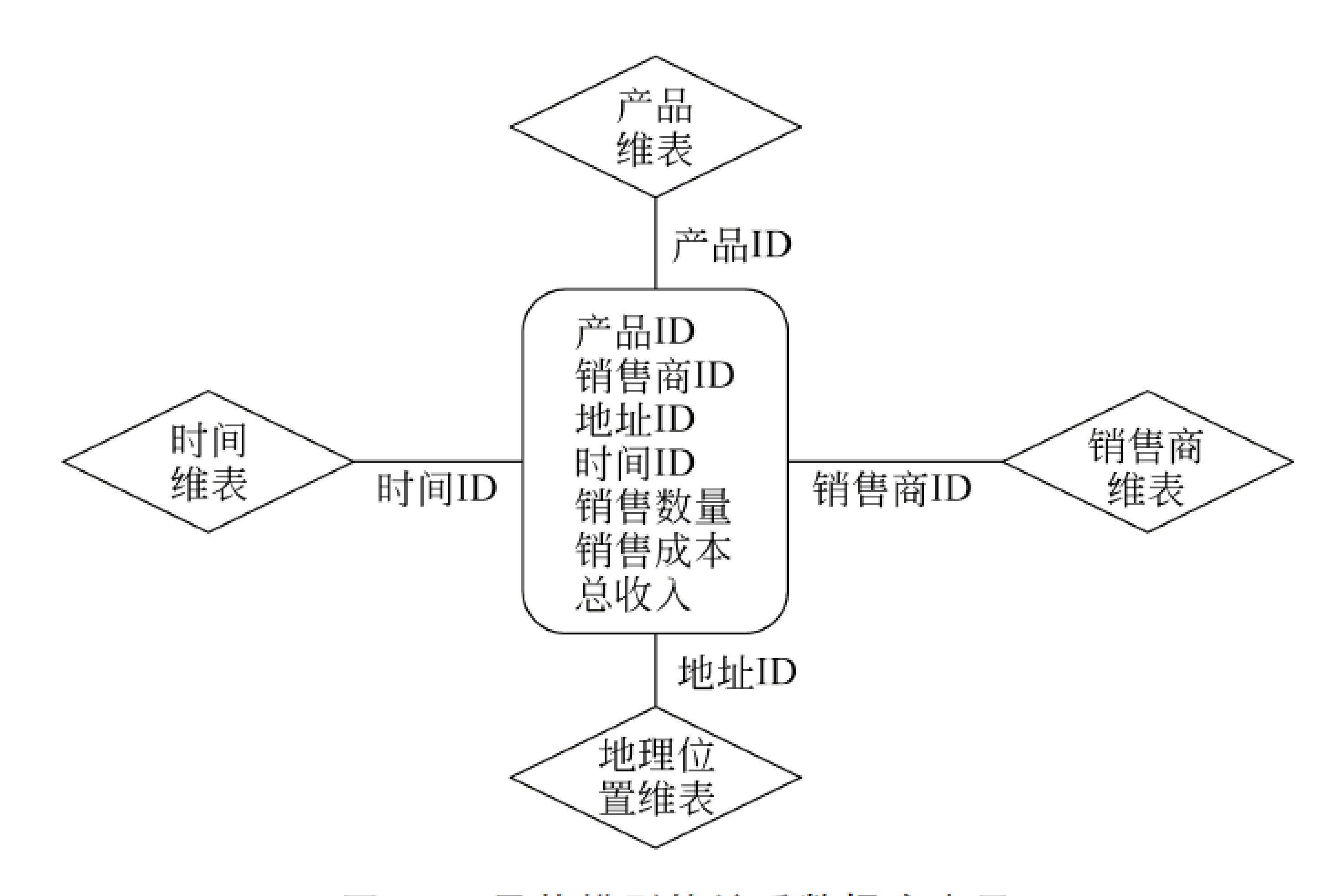


图 3-7 星状模型的关系数据库表示

（2）雪花状模式。雪花状模式是星状模式的变种，其中某些维是规范化的，因而把数据进一步分解到附加表中，结果模式图就会形成类似于雪花的形状。例如，图 3-8 表示的是一个雪花模式。

雪花状模式和星状模式的区别在于，雪花状的维表可能是规范化形式，以便减少冗余。这种表易于维护并节省存储空间。然而，与巨大的事实表相比，这种空间的节省可以忽略。此外，由于执行查询需要更多的连接操作，雪花状结构可能降低浏览的性能。系统的性能可能受到相对影响。因此，尽管雪花状模式减少了冗余，但是在数据仓库设计中，雪花模式不如星状模型流行。

2. ROLAP 的创建步骤

ROLAP 的创建和 MOLAP 的创建一样需要进行选择、确定、构造、定义，然后还需要完成数据管理、元数据存储、应用工具构造等操作。

（1）数据管理。为了合理有效地进行关系数据库的管理，需要在数据库中添加合适的聚集数据和概括数据，将较大的数据库分解成可管理的部分，添加生成索引和位图索引来提高 ROLAP 的处理效率。

（2）元数据存储。ROLAP 的应用主要依赖元数据的生成和存储。

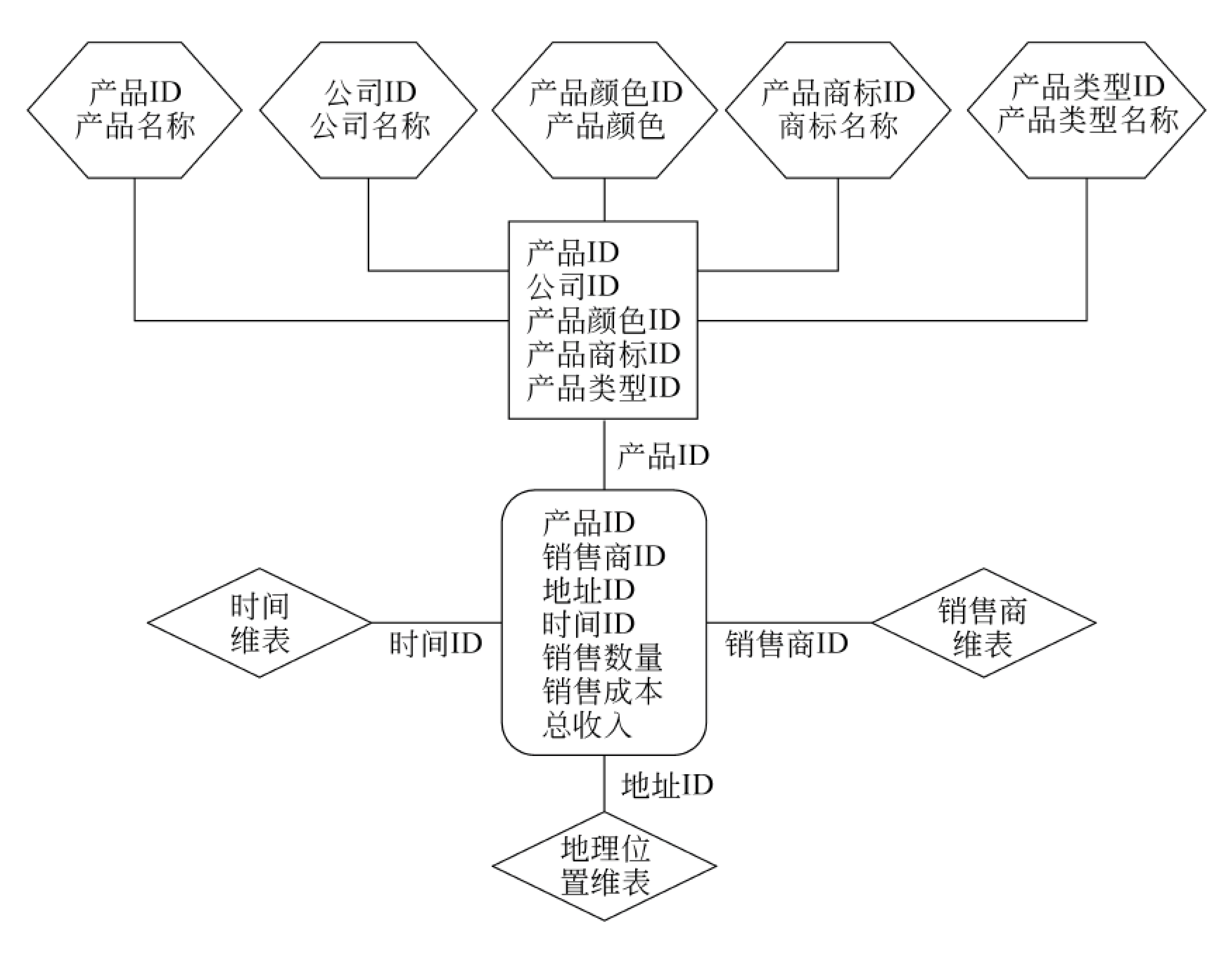


图 3-8 雪花状模式的关系数据库表示

(3) 应用工具构造。需要利用数据库的应用视图或维视图构造客户工具。数据库可以利用查询结果进行多维操作,实现计算、公式、数据到应用的转化,并可以将结果及时地反映给用户,在进行进一步处理后,再显示给客户。

3. ROLAP 的功能

(1) 细节剖析。允许用户在 ROLAP 上进行数据的聚集、概括分级、分解和剖析细节,并且可以对其子集进行个案的分析。

(2) 数据的备份和安全功能。用户不仅可以对数据库进行数据的备份和安全管理,而且还可以由数据管理员进行增强性的控制。

(3) 数据的商业视图。商业视图的设计是基于维模型的,可以通过 ROLAP 将星状模型、雪花状模型和混合模型转化为商业视图。

(4) 元数据导航功能。ROLAP 可以对全局数据库新生的元数据进行合理的导航。

(5) 维层次支持。ROLAP 需要能够提供维层次操作的支持,能够实现维层次与关系数据存储的转化与管理。

(6) 模型的自定义。ROLAP 允许用户对分析模型进行自定义,根据决策分析的需要选择不同的计算、统计和各种分析模型。

ROLAP 的主要特点是它的灵活性强,用户可以动态定义统计或计算方式。ROLAP 的缺点是它对用户的分析请求处理时间要比 MOLAP 长。

3.3.3 MOLAP 和 ROLAP 的比较

多维联机分析处理和关系联机分析处理的比较分析如表 3-3 所示。

表 3-3　ROLAP 和 MOLAP 的比较

ROLAP	MOLAP
沿用现有的关系数据库的技术	专为 OLAP 所设计
响应速度比 MOLAP 慢	性能好、响应速度快
数据装载速度快	数据装载速度慢
存储空间耗费小，维数没有限制	需要进行预计算，可能导致数据爆炸，维数有限；无法支持维的动态变化
借用关系数据库存储数据，没有文件大小限制	受操作系统平台中文件大小的限制
可以通过 SQL 实现详细数据与概要数据的存储	缺乏数据模型和数据访问的标准
不支持有关预计算的读写操作 SQL 无法完成部分计算 无法完成多行的计算 无法完成维之间的计算	支持高性能的决策支持计算 复杂的跨维计算 多用户的读写操作 行级的计算
维护困难	管理简便

3.3.4　混合型联机分析处理

由于 MOLAP 与 ROLAP 有着各自不同的优缺点，且它们的结构也不同，这给分析人员设计 OLAP 结构时提出了难题，他们必须在两种结构之间进行筛选，为此一个新的 OLAP 结构——混合型 OLAP 被提出。在 HOLAP 中，原始数据和 ROLAP 一样存储在原来的关系数据库中，而聚合数据则以多维的形式存储。

HOLAP 结构不是 MOLAP 与 ROLAP 结构的简单组合，而是这两种结构技术优点的有机结合，能满足用户各种复杂的分析请求。一个真正的 HOLAP 系统应能遵循以下几个准则。

(1) 维数能够被动态更新，一个真正的 HOLAP 不但可以提供对数据的实时存取，还可以根据不断变化的结构对维数进行更新。

(2) 可根据关系数据库管理系统的元数据产生多维视图。

(3) 可以快速地存取各种级别的汇总数据。

(4) 可适应大数据量数据的分析。

(5) 可以方便地对计算和汇总算法进行维护和修改。

3.4　OLAP 的衡量标准

1993 年 E. F. Codd 提出了关于 OLAP 的 12 条标准，其目的是希望能加深对 OLAP 的理解。事实上，这些标准已经成为 OLAP 工具所应该具有的关键特性的最小描述。尽管 Codd 提出的 12 个准则也需要不断完善，但现阶段仍是评价和购买 OLAP 产品的参考标准。这些标准主要包括：

准则 1　OLAP 模型必须提供多维概念视图。

从用户分析员的角度来看，整个企业的视图本质上是多维的，因此 OLAP 的概念模型

也应该是多维的。企业决策分析的目的不同，决定了分析和衡量企业的数据总是从不同角度来进行的，所以企业数据空间本身就是多维的。一个 OLAP 产品作为分析的工具，应该提供直观的多维分析模型进行分析、设计和维内到维间的计算。这种多维模型可以使最终分析以比单一维模型更简单、直观的方式操纵多维数据。通过对多维数据模型进行切片、切块和维旋转就可以轻松地完成传统的操作方法必须用较长的时间和极大的代价才能完成的工作。

准则 2　透明性准则。

透明性原则包括两层含义：首先，OLAP 在体系结构中的位置对用户是透明的。OLAP 应处于一个真正的开放系统结构中，允许分析工具嵌入到分析员指定的任何位置而不影响嵌入工具的性能。这对保持用户现有的效率，保证良好的性能至关重要。同时必须保证 OLAP 的嵌入不会引入和增加任何复杂性；其次，OLAP 的数据源对用户的需求是只需使用熟悉的查询工具进行查询，而不必关心输入 OLAP 工具的数据是来自于同质还是异质的企业数据源。

准则 3　存取能力准则。

OLAP 系统不仅能进行开放的存取，而且还能提供高效的存取策略。OLAP 用户分析员能在公共概念视图的基础上对关系数据库和外部存储的数据进行分析。要实现这些功能，就要求 OLAP 能将自己的概念视图映射到异质的数据存储上，能访问数据并执行所需的转换，从而提供单一、完整的用户视图。另外 OLAP 系统应提供高效的存取策略，应使系统只存取与指定分析有关的数据，避免多余的数据存取。

准则 4　稳定的报表性能。

当数据维数和数据综合层次增加时，提供给最终分析员的报表能力和响应速度不应该有明显的降低和减慢，这对维护 OLAP 产品的易用性和低复杂性至关重要。即便是企业模型改变时，关键数据的计算方法也无须更改。也就是说，OLAP 系统模型对企业模型应该具有“鲁棒”性。只有做到这一点，OLAP 工具提供的数据报表和所做的预测分析结果才是可信的。

准则 5　客户/服务器体系结构。

OLAP 是建立在客户/服务器体系结构上的。这要求它的多维数据服务器能被不同的应用工具访问到。服务器端智能地以最小的代价完成同多种服务器之间的挂接任务；服务器端必须完成分散的企业数据的逻辑模式和物理模式之间的映射，并确保它们的一致性，从而保证透明性和建立统一的公共概念模式、逻辑模式和物理模式。客户端负责应用逻辑及用户界面。

准则 6　维的等同性准则。

每一数据维在数据结构和操作能力上都是等同的。系统可以将附加的操作能力授给所选维，但必须保证该操作能力可以授给任意其他维，即要求维上的操作是公共的。

准则 7　动态的稀疏矩阵处理准则。

OLAP 工具的物理模型必须充分适应指定的分析模型，提供“最优”的稀疏矩阵处理，这是 OLAP 工具应遵循的最重要的准则之一。该准则包括两层含义：第一，对任意给定的稀疏矩阵，存在且仅存在一个最优的物理视图，该最优视图能提供最大的内存效率和矩阵处理能力；稀疏度是数据分布的一个特征，不能适应数据集合的数据分布，将会导致快速、高效操作失败；第二，OLAP 工具的基本物理数据单元可配置给可能出现的维的子集。同时，还要提供动态可变的访问方法并包含多种存取机制。

准则 8　多用户支持能力准则。

多个用户分析员可以同时工作于同一分析模型上或是可以在同一企业数据上建立不同的分析模型。OLAP 工具必须提供并发访问、数据完整性及安全性机制。

准则 9　非受限的跨维操作。

多维数据之间存在固有的层次关系，这就要求 OLAP 工具能自己推导出而不是最终用户明确定义出相关的计算。对于无法从固有关系中得出的计算，要求系统提供计算完备的语言来定义各类计算公式。

准则 10　直观的数据操纵。

这一准则要求数据操纵直观易懂，综合路径重定位、向上综合、向下挖掘和其他操作都可以通过直观、方便的点击操作完成。

准则 11　灵活的报表生成。

报表必须从各种可能的方面显示出从数据模型中综合出的数据和信息，充分反映数据分析模型的多维特征。

准则 12　非受限维与聚集层次。

OLAP 工具的维数应不小于 15 维，用户分析员可以在任意给定的综合路径上建立任意多个聚集层次。

3.5　基于 SQL Server 2005 的 OLAP 实现

银行在其业务中要面对的数据是海量的，工作人员的工作量是巨大的，故准确定位工作重点的必要性日益突出。本节中，将对某商业银行在信贷业务中产生的大量数据进行分析，简单介绍在银行信贷业务中如何应用联机分析处理技术在统揽全局、总体把握的基础上对海量数据进行筛选、定位和深入分析。其目的是借助 SQL Server 2005 软件，通过联机分析处理技术，从时间、分行代码、客户代码、余额、损失等不同维度进行分析，以帮助银行工作人员准确定位工作重点。

1. 启动 SQL Server 2005

(1) 执行"开始"|"程序"|SQL Server Management Studio 菜单命令，出现如图 3-9 所



图 3-9　"连接到服务器"对话框

示的“连接到服务器”对话框。

（2）单击“连接”按钮，如图 3-10 所示。

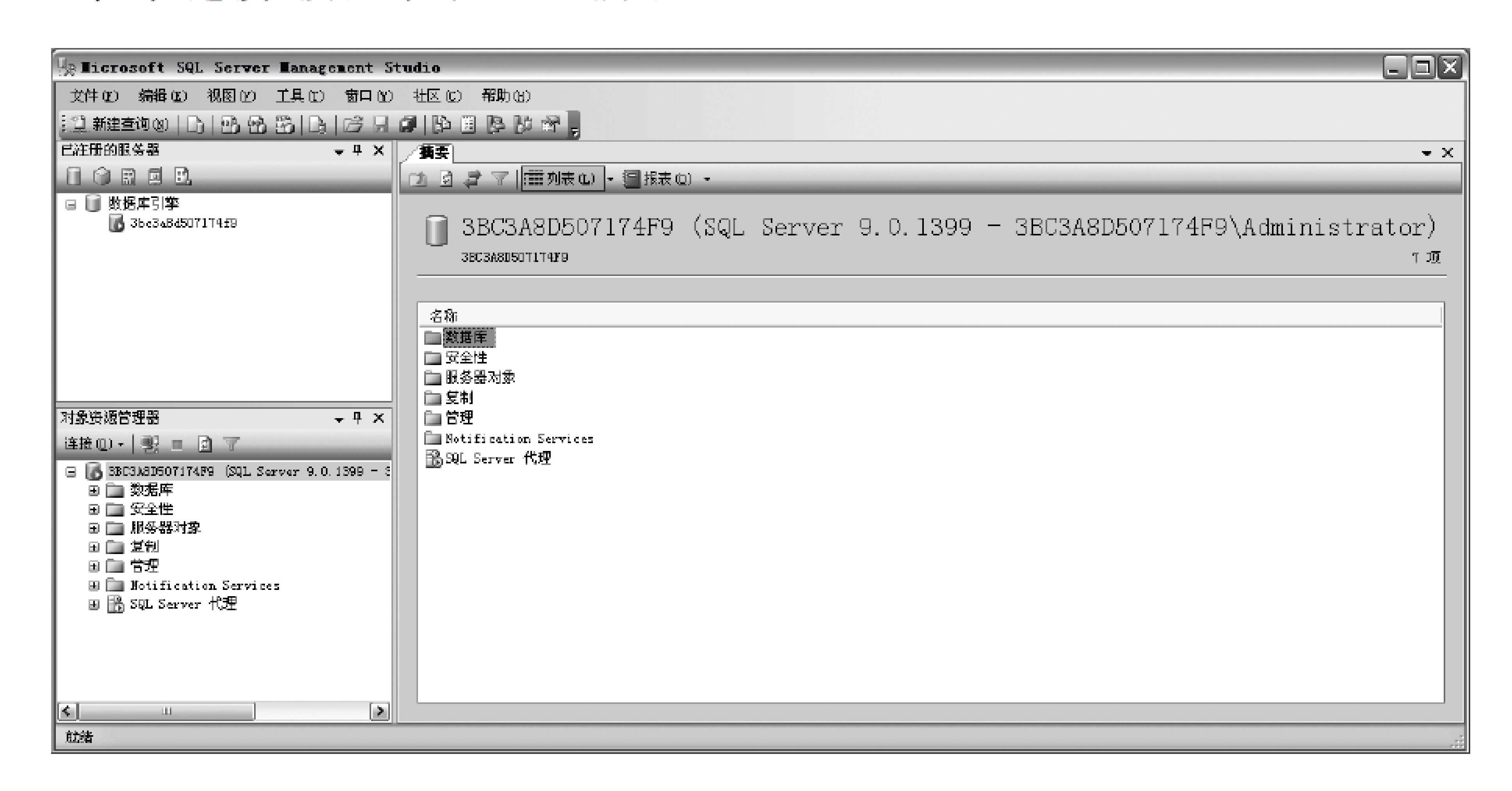


图 3-10　Microsoft SQL Server Management Studio

2. 新建数据仓库

在构建多维数据集之前，需要新建一个数据仓库，以存放数据源、多维数据集、共享数据维度、挖掘模型和数据库角色等对象。

（1）右击“数据库”选项，弹出“新建数据库”对话框，在“数据库名称”一栏填写“商业银行”如图 3-11 所示。

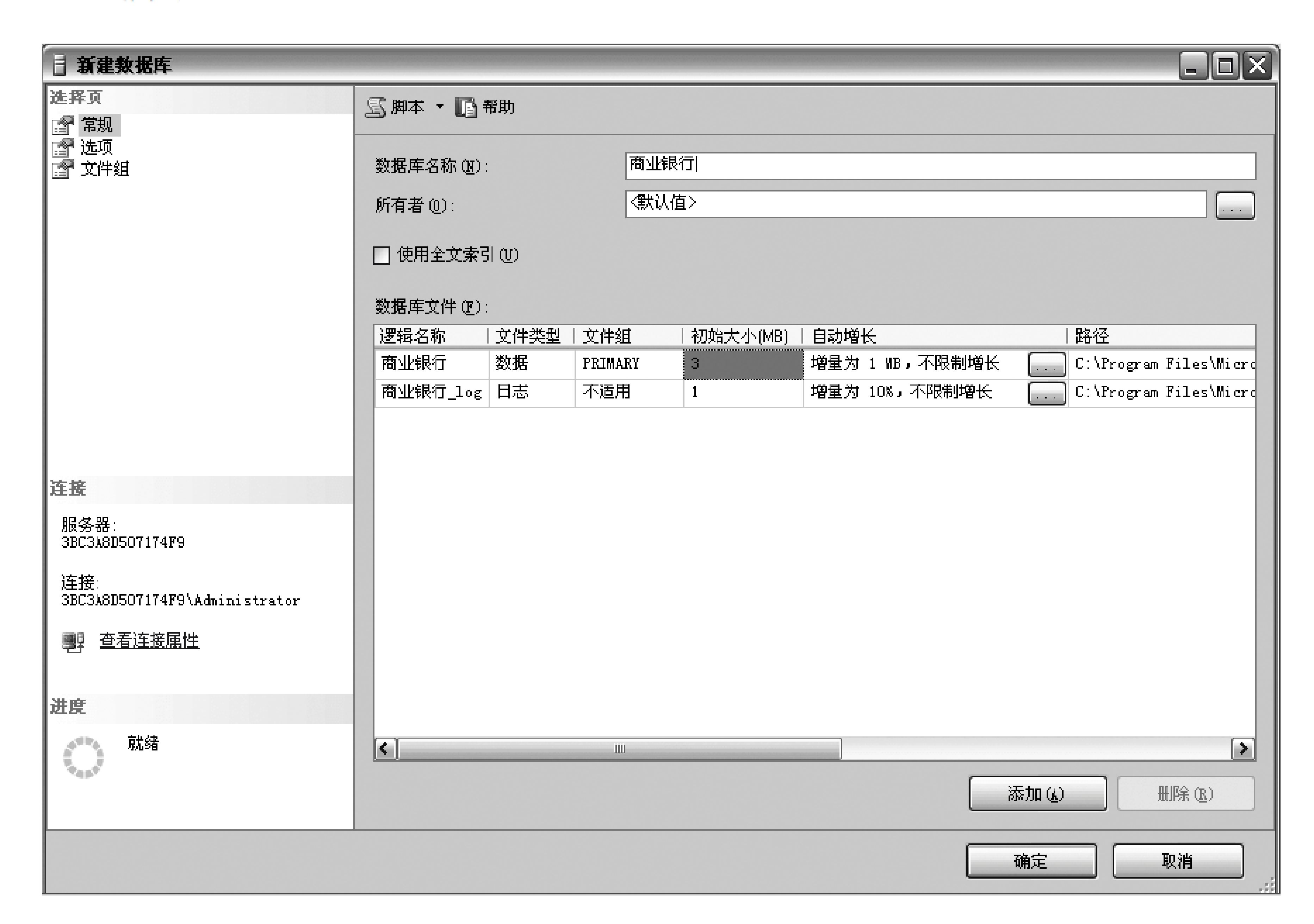


图 3-11　“新建数据库”对话框

（2）单击“确定”按钮，在“对象资源管理器”中的“数据库”的下拉菜单中找到“商业银行”，右击后选择“任务”|“导入数据”命令，如图 3-12 所示。



图 3-12 “导入数据”菜单项

(3) 在弹出的如图 3-13 所示的对话框中单击“下一步”按钮,在“数据源”栏中适当的选择,并在“数据库”栏中选择“商业银行”,如图 3-14 所示。

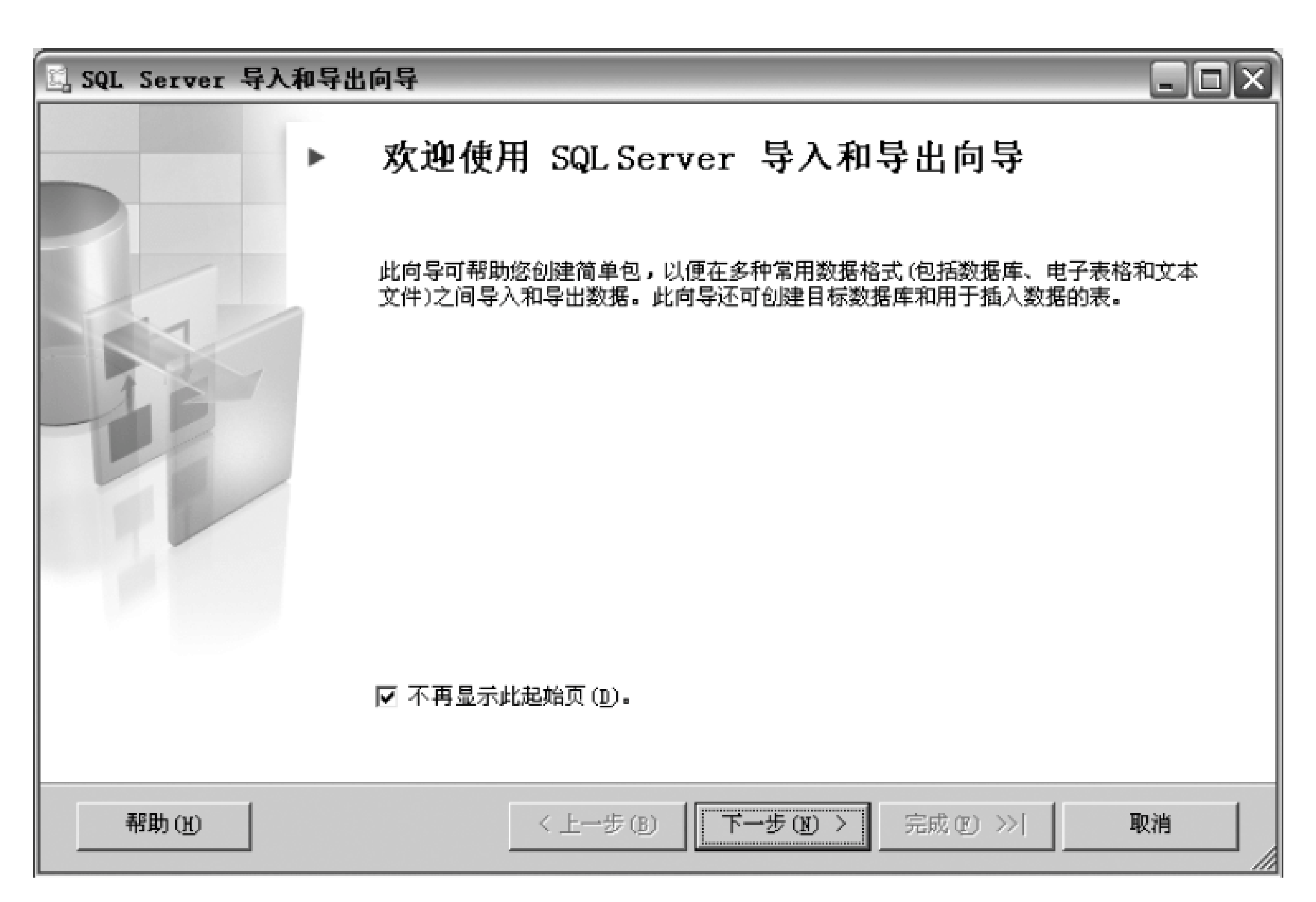


图 3-13 导入导出向导

(4) 单击“浏览”按钮找到所要导入的数据,如图 3-15 所示。

(5) 单击“下一步”按钮弹出如图 3-16 所示对话框,将“贷款余额表”、“分行代码表”、“客户基本情况表”全部选中。

(6) 单击“下一步”按钮,直至完成,进入到如图 3-17 所示的验证窗口。

3. 新建数据源

(1) 执行“开始”|Microsoft Visual Studio 菜单命令,如图 3-18 所示。

(2) 右击“数据源”,选择“新建”命令,在弹出的对话框中单击“下一步”按钮,如图 3-19 所示。

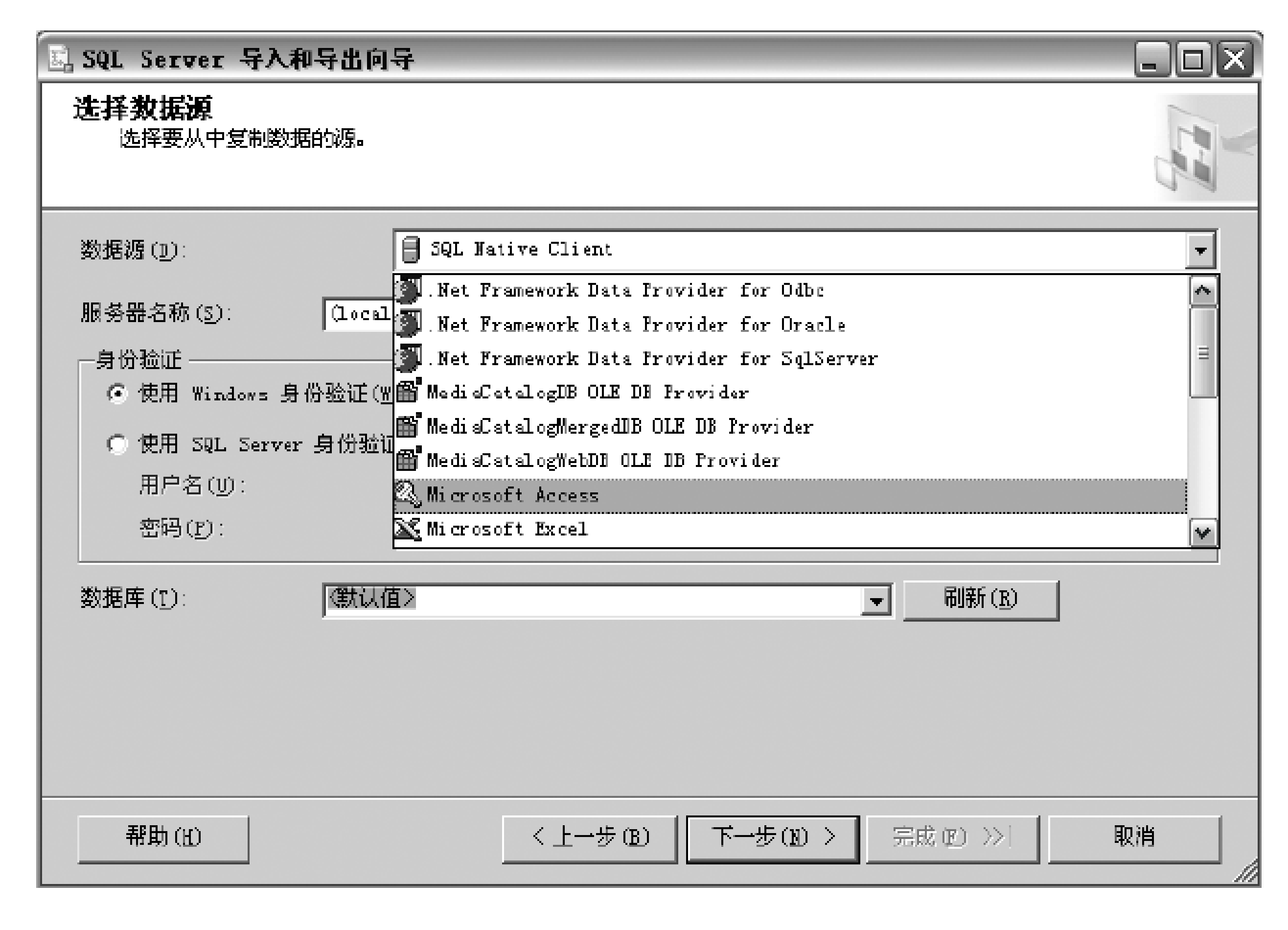


图 3-14　数据源和数据库的选择

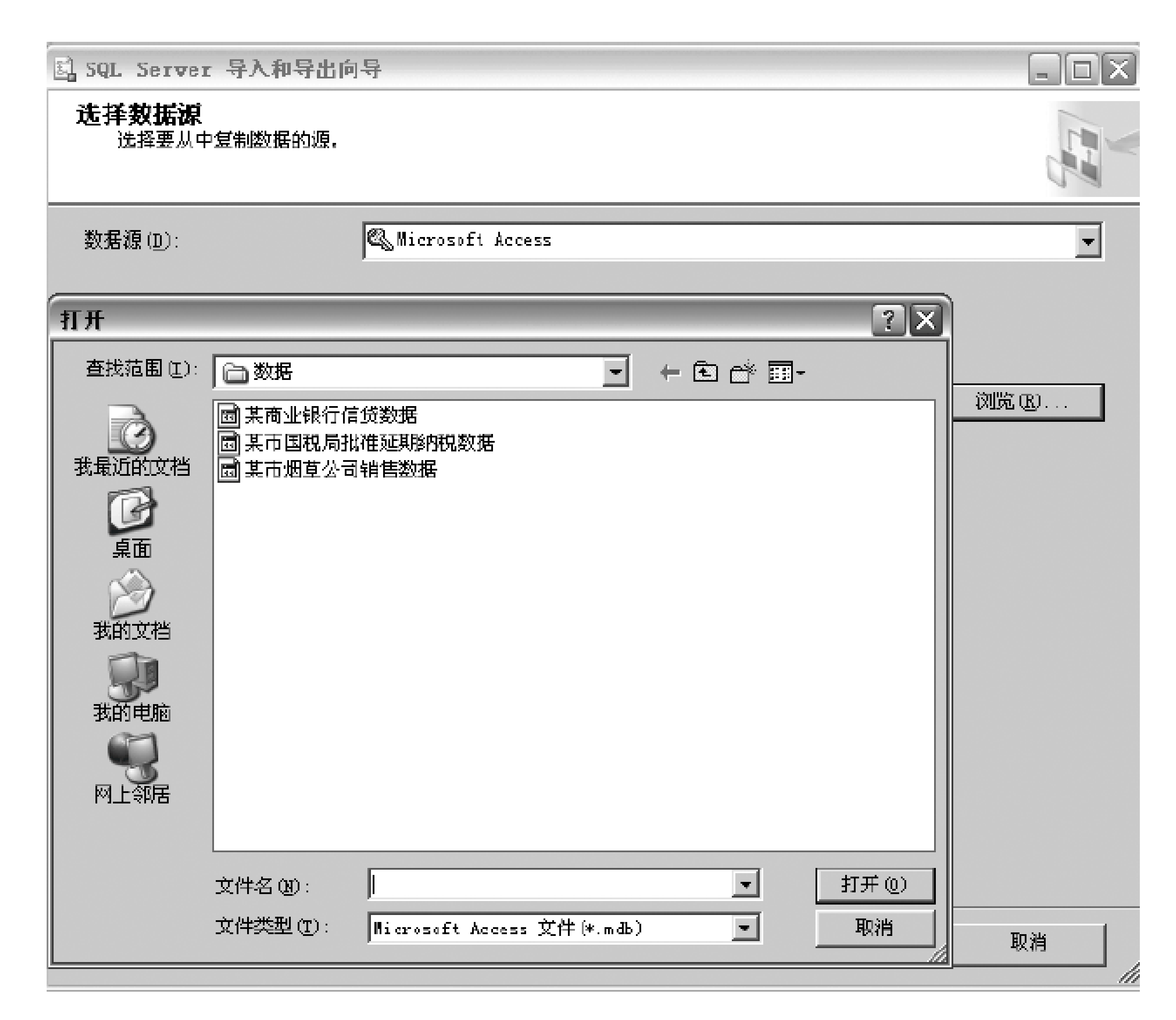


图 3-15　选择导入



图 3-16　选择数据表和数据源



图 3-17　验证窗口

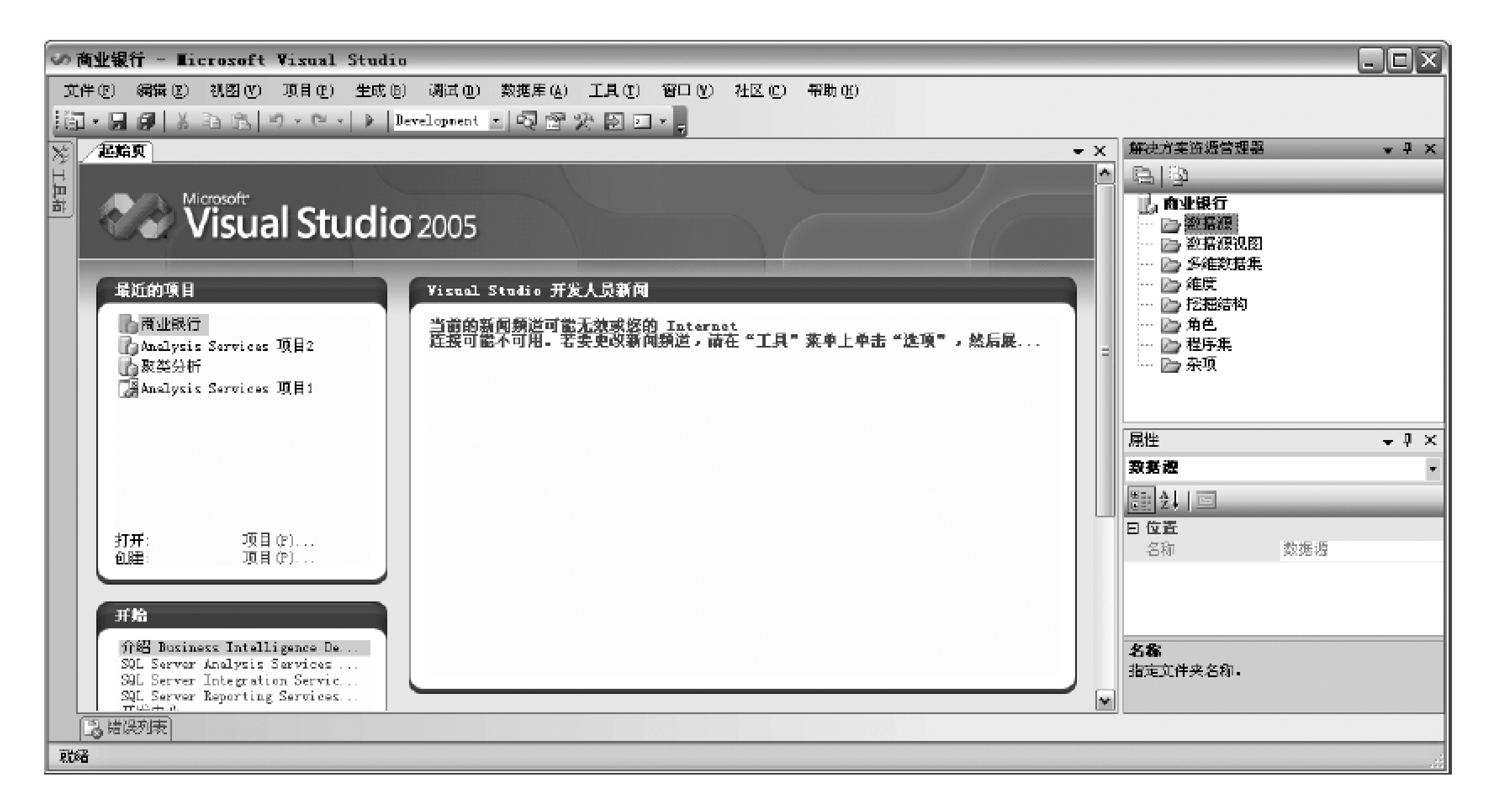


图 3-18　Microsoft Visual Studio



图 3-19　数据源向导

（3）单击“新建”按钮进入连接管理器，输入服务器名称，选择数据库名，如图 3-20 所示。



图 3-20　资源管理器

（4）单击“确定”按钮，进行下一步操作，选择“使用服务账户”单选按钮，如图 3-21 所示。



图 3-21　定义数据源

4．建立数据源视图

在图 3-18 所示窗口中右击“数据源视图”，选择“新建”命令，在弹出的对话框中单击“下一步”按钮，最后单击“完成”按钮，如图 3-22～图 3-26 所示。

“数据源视图”创建完成时，如图 3-27 所示。



图 3-22　数据源视图向导



图 3-23　选择数据源

数据源视图向导
名称匹配
找不到外键。可以在匹配的列上创建逻辑关系。
通过匹配列创建逻辑关系(L)
外键匹配
与主键同名(S)
与目标表同名(M)
目标表名 + 主键名(D)
说明和示例:
源外键列名与目标表的主键列名相同。例如：
Order.CustomerID 与 Customer.CustomerID
< 上一步(B)
下一步(N) >
完成(F) >>|
取消

图 3-24　名称匹配



图 3-25 选择表和视图



图 3-26 完成向导



图 3-27 数据源视图

5. 数据浏览

(1) 右击“贷款余额表”,在弹出的快捷菜单中选择“浏览数据”命令,如图 3-28 所示。对已获得的数据进行浏览,原始数据表如图 3-29 所示。

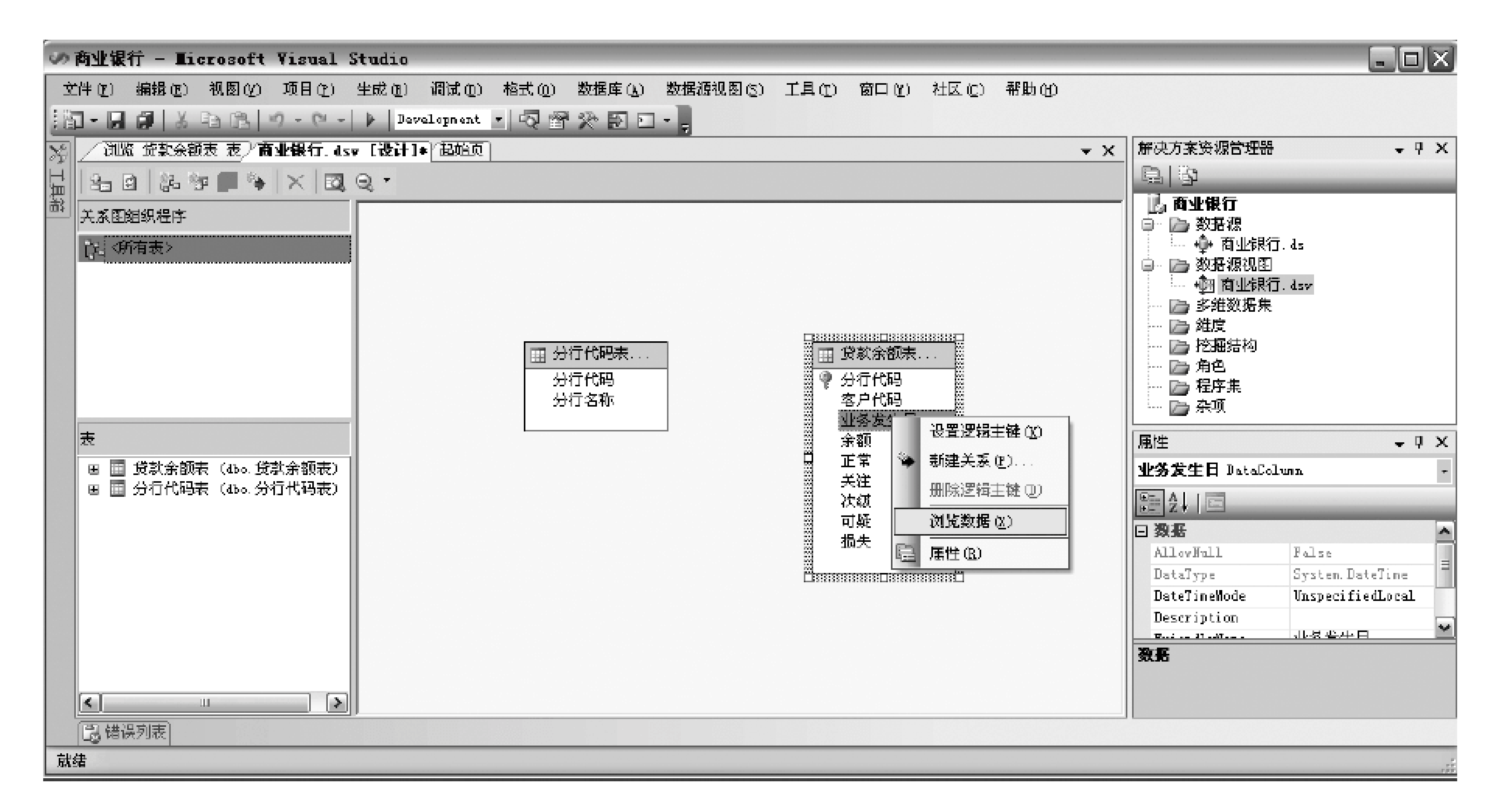


图 3-28 数据浏览

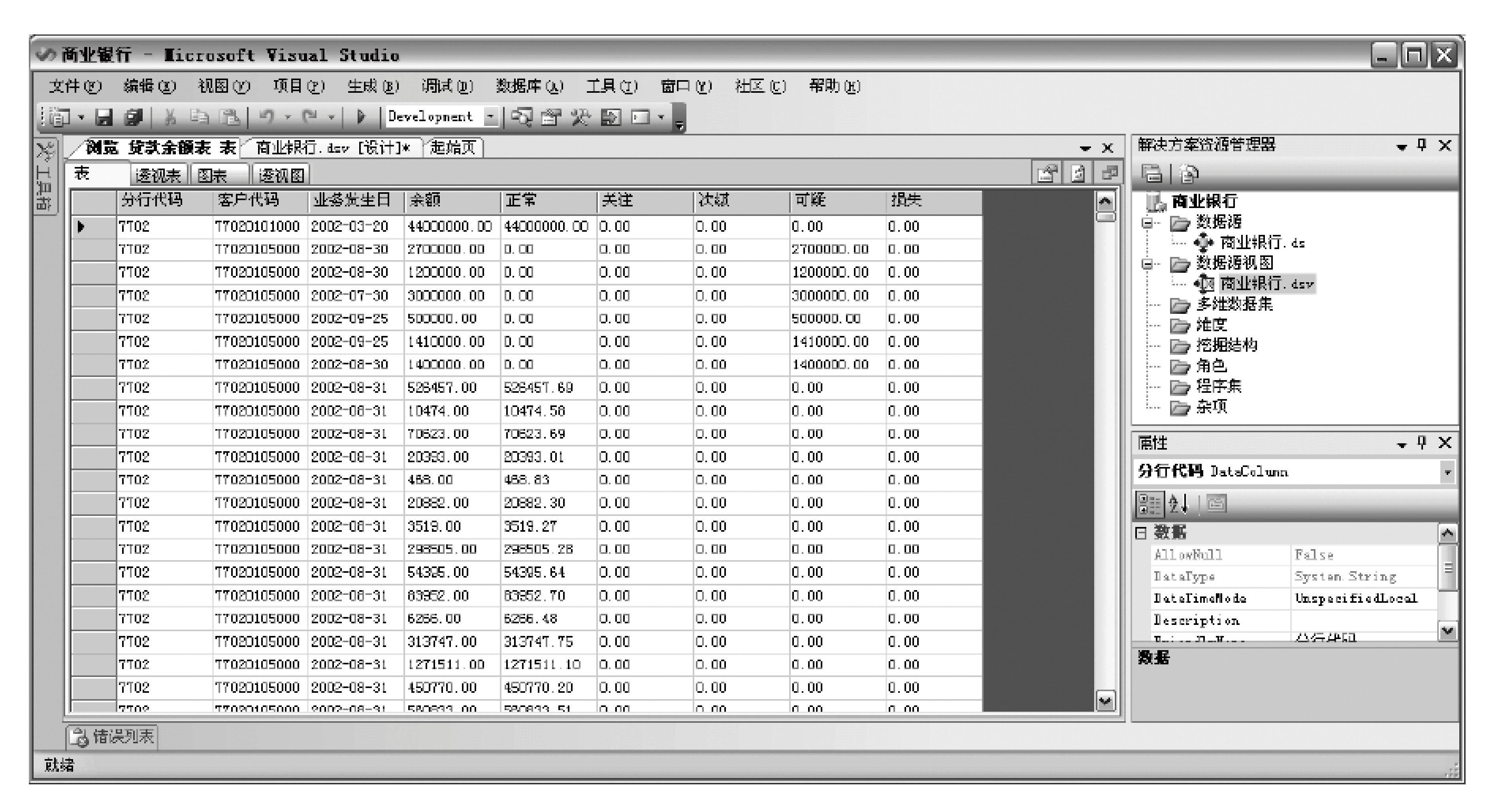


图 3-29 原始数据表

(2) 选择“透视表”,可选择需要的维度对原始数据进行分析,如图 3-30 所示。在此选择“分行代码”和“业务发生日”两个维度进行分析。

(3) 选择“图表”,可选择需要的维度对原始数据进行分析,如图 3-31 所示。选择分行代码、客户代码、次级、可疑等维度进行分析。

(4) 选择“透视图”,可选择需要的维度对原始数据进行分析,如图 3-32 所示。在透视表中可选择需要的维度(可疑、关注、余额、业务发生日)进行分析。

(5) 把“损失”和“业务发生日”两个维度拖入表中,如图 3-33 所示。

(6) 由于数据库中包含了整整一年的业务记录,数据量比较大,所以把“业务发生日”按月份拖入,可以看出六月份损失最多达到 124987740,所以银行工作人员应把工作重点放到

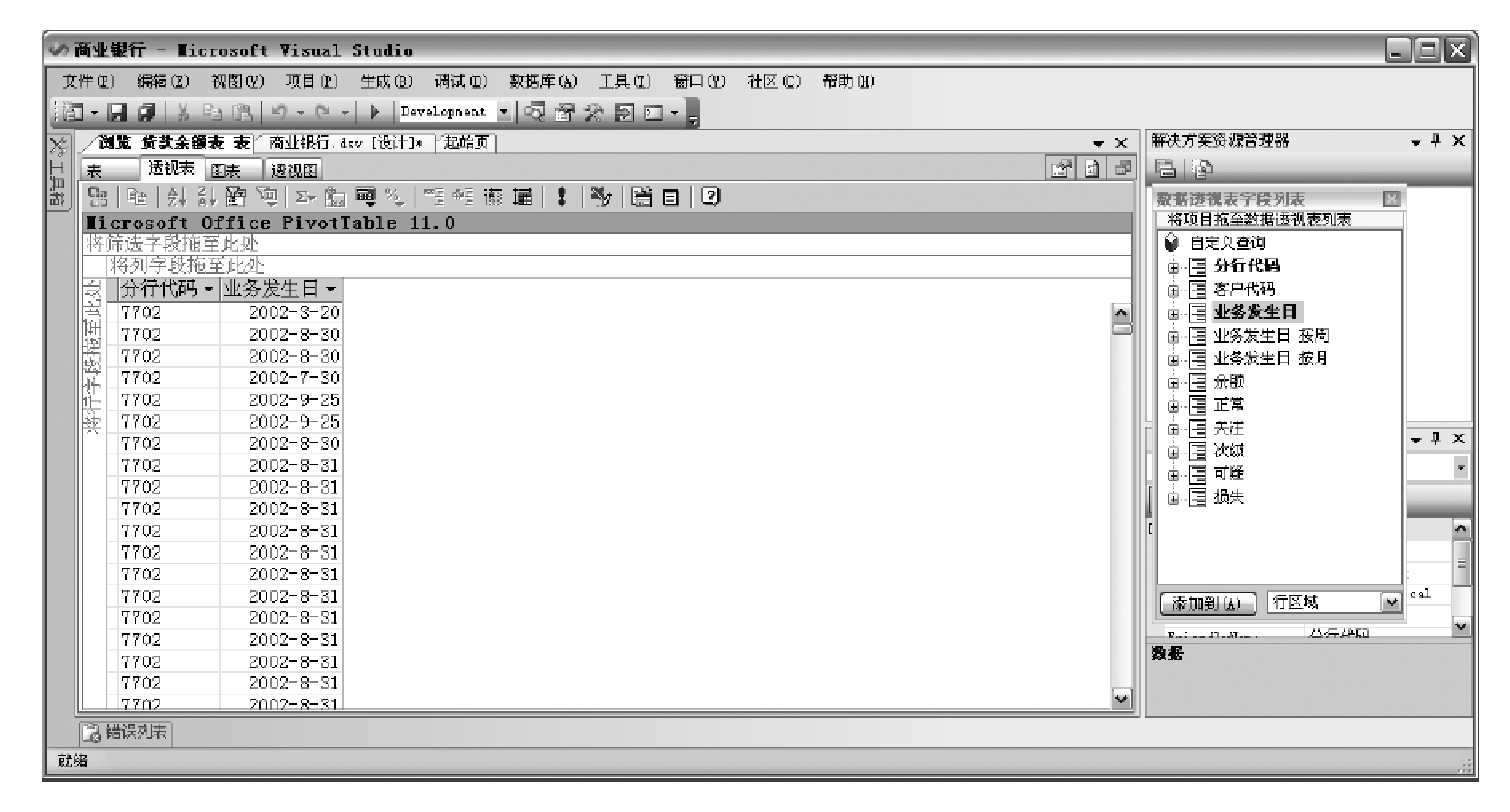


图 3-30　从透视表分析

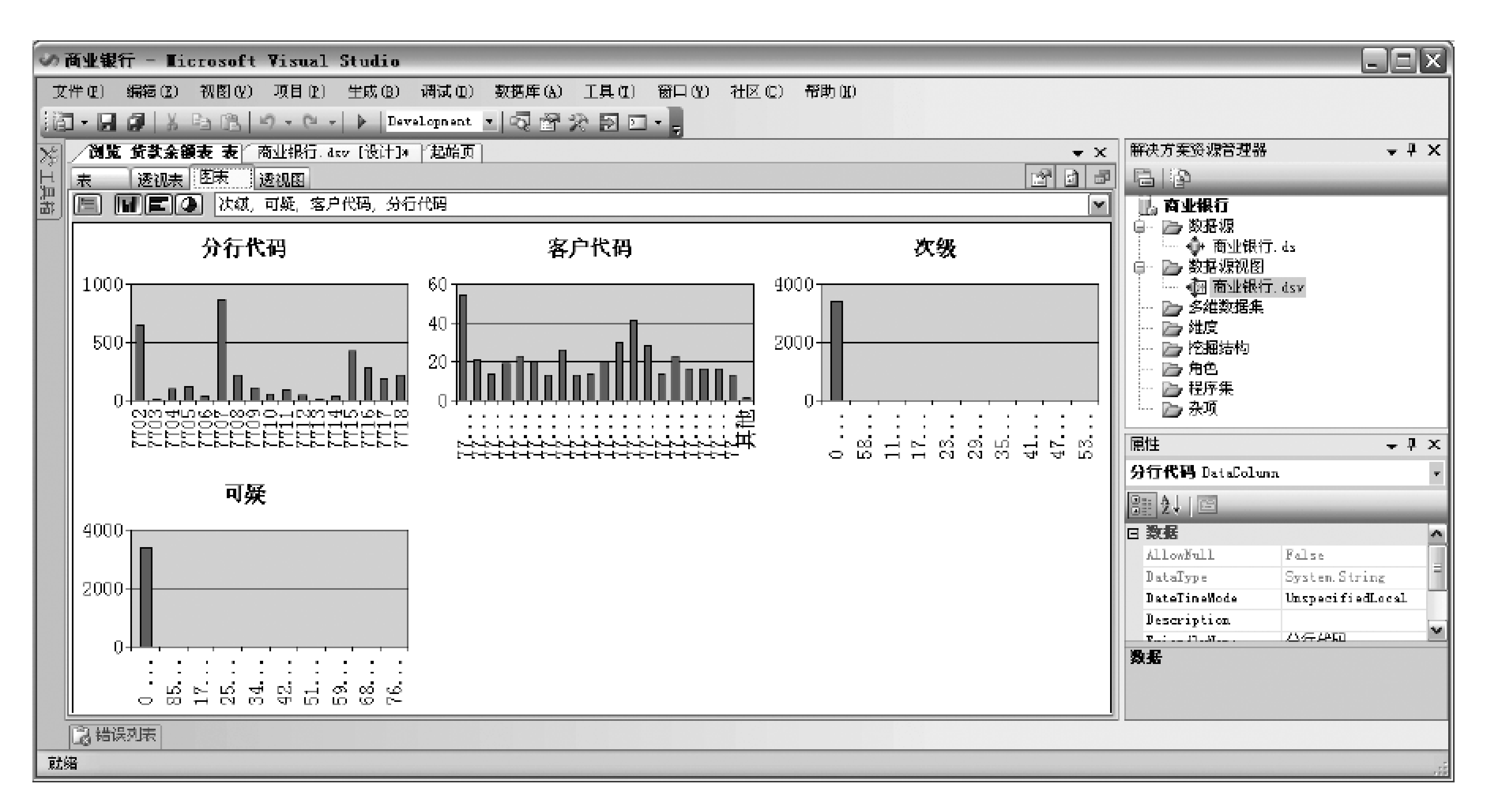


图 3-31　从图标分析

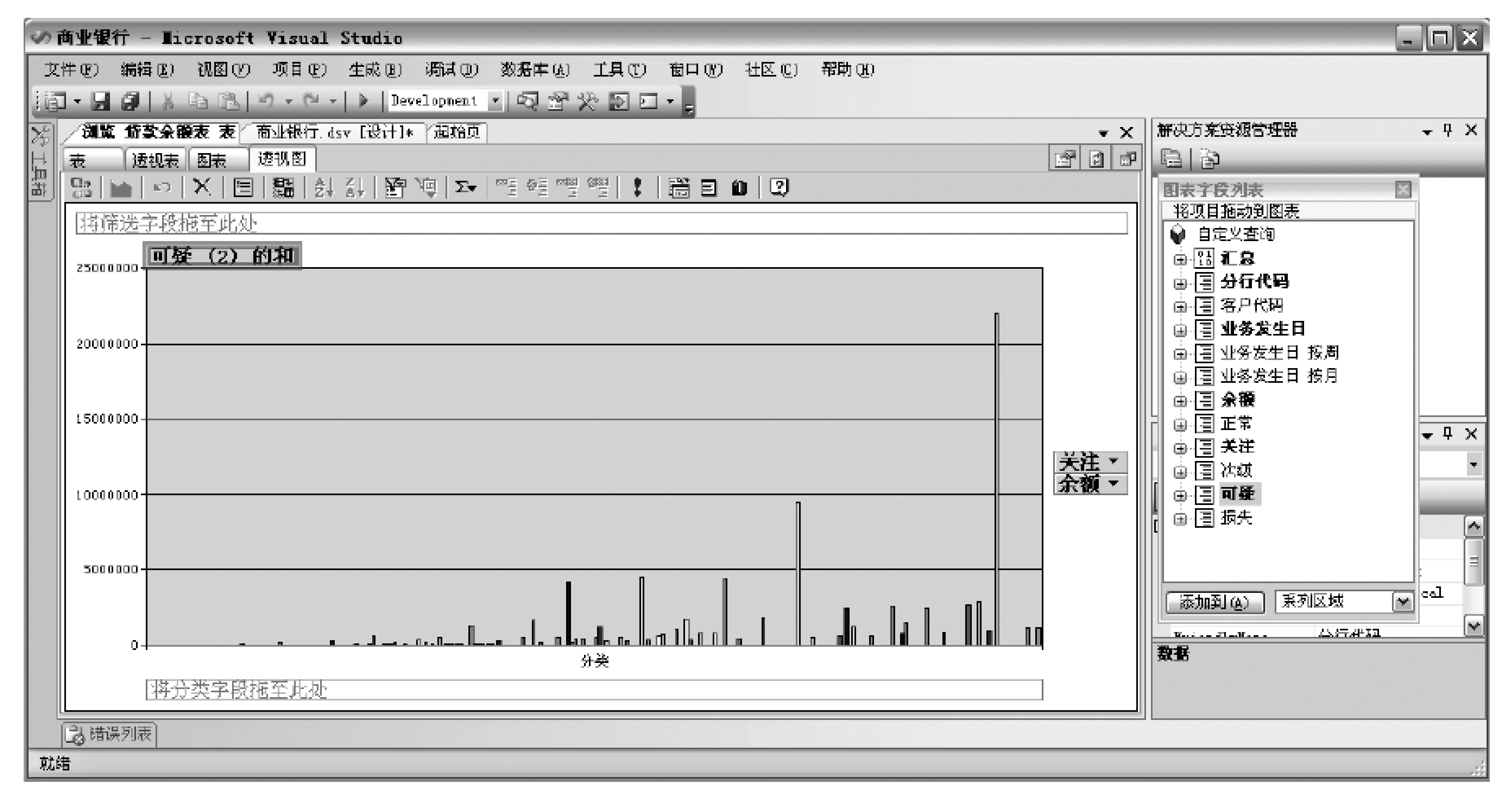


图 3-32　从透视图分析

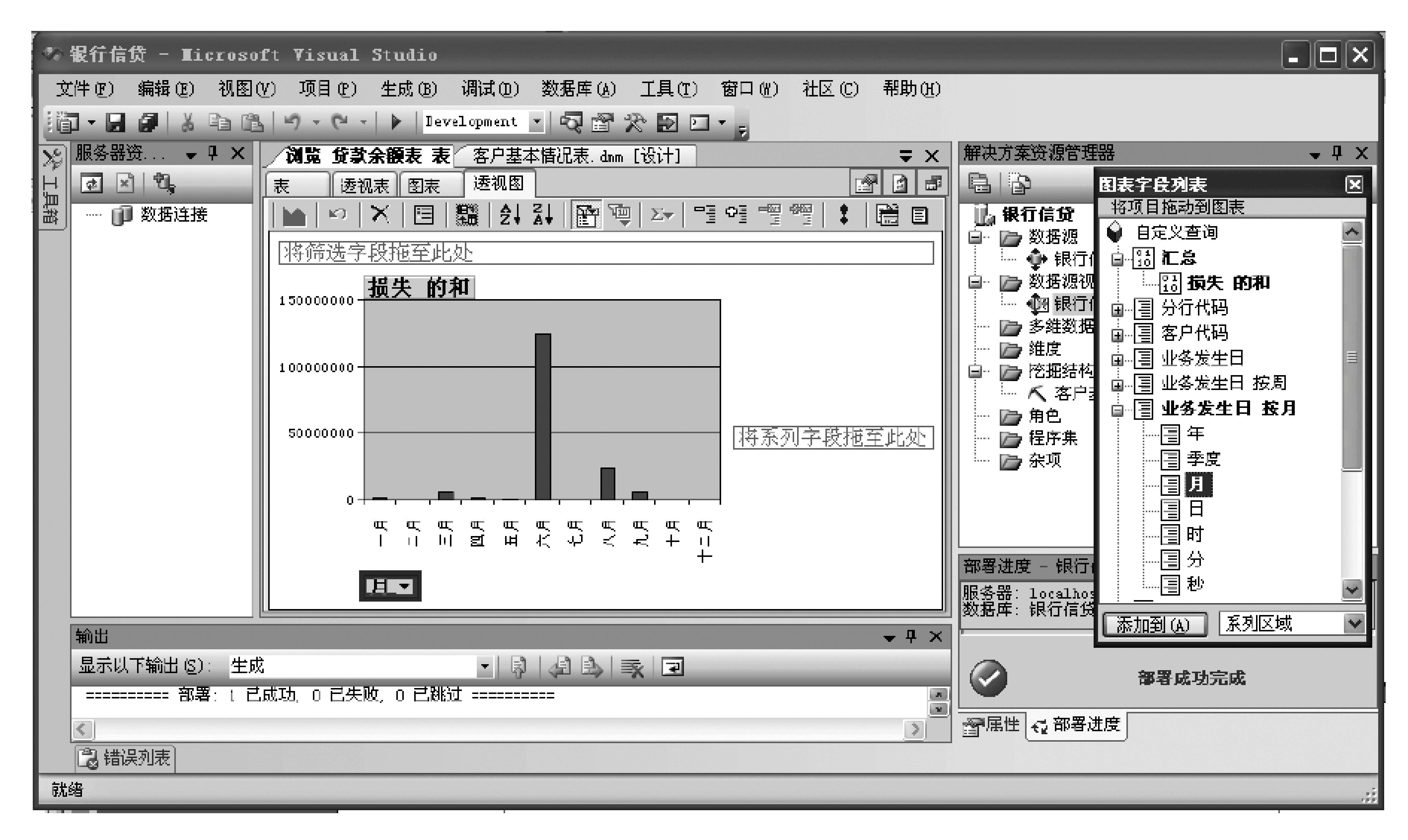


图 3-33　月和损失

六月份的相关数据上，为了进一步缩小范围，细化到六月份每天的损失状况，所以可把维度“日”拖入表中，如图 3-34 所示。

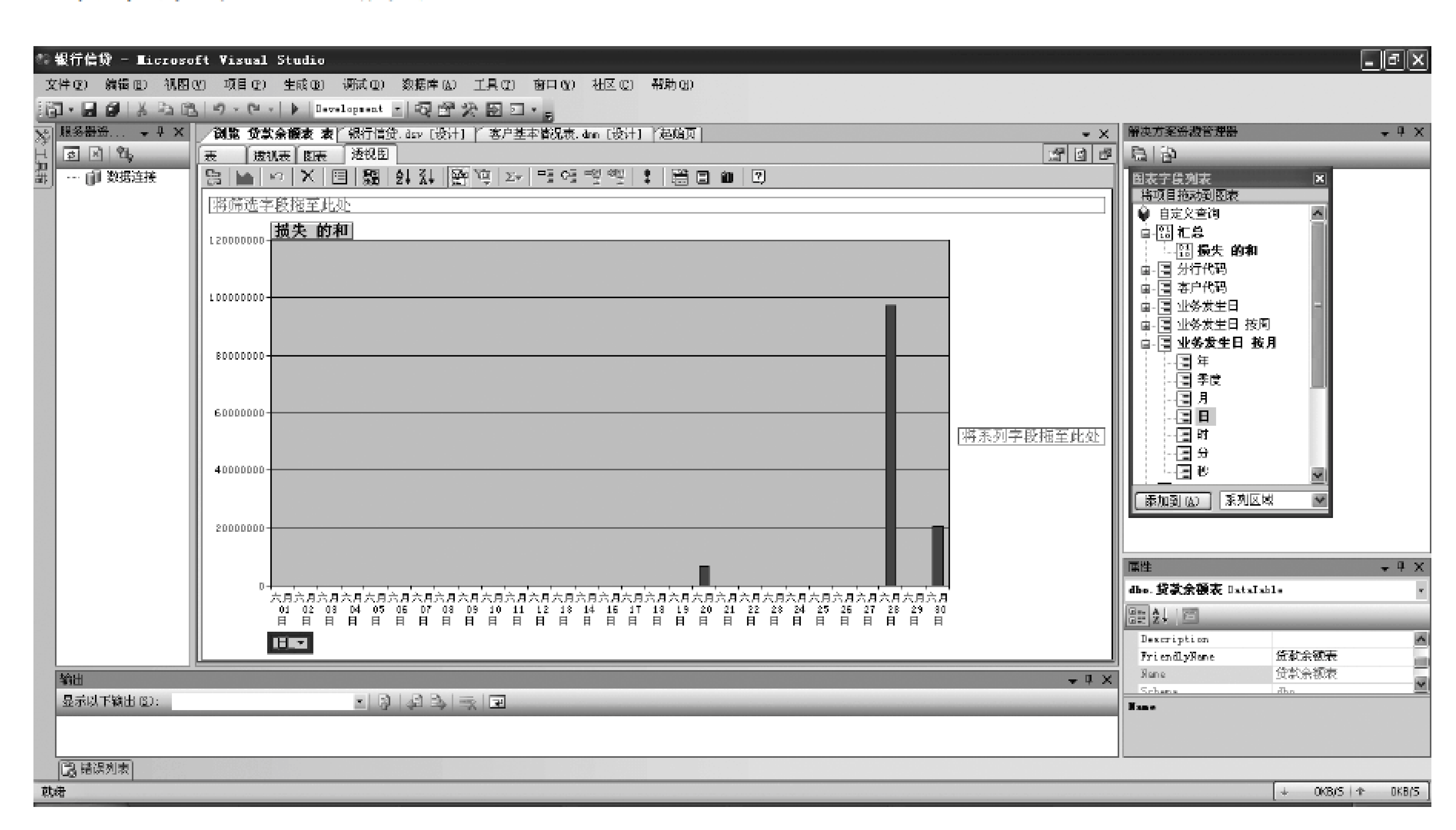


图 3-34　日和损失

（7）从图 3-34 中可以看出损失最多的是六月 28 日、30 日这两天，故可以把工作重点进一步缩小到 28 日，再把“分行代码”维度拖入表中，具体分析是哪家分行损失最多，如图 3-35 所示。

（8）可以看出代码为 7707 的分行损失最多，所以可把 7707 分行作为分析重点，然后把“损失”、“日”、“分行代码”三个维度同时拖入表中，进行三维分析，如图 3-36 所示。

（9）从图 3-36 可以得出 7707 分行在六月 28 日的损失额为 94438000，在此可确定其为突破口，为了具体研究 7707 在 6 月 28 日的业务情况，可把维度换为“分行代码”、“客户代码”、“日”，如图 3-37 所示。

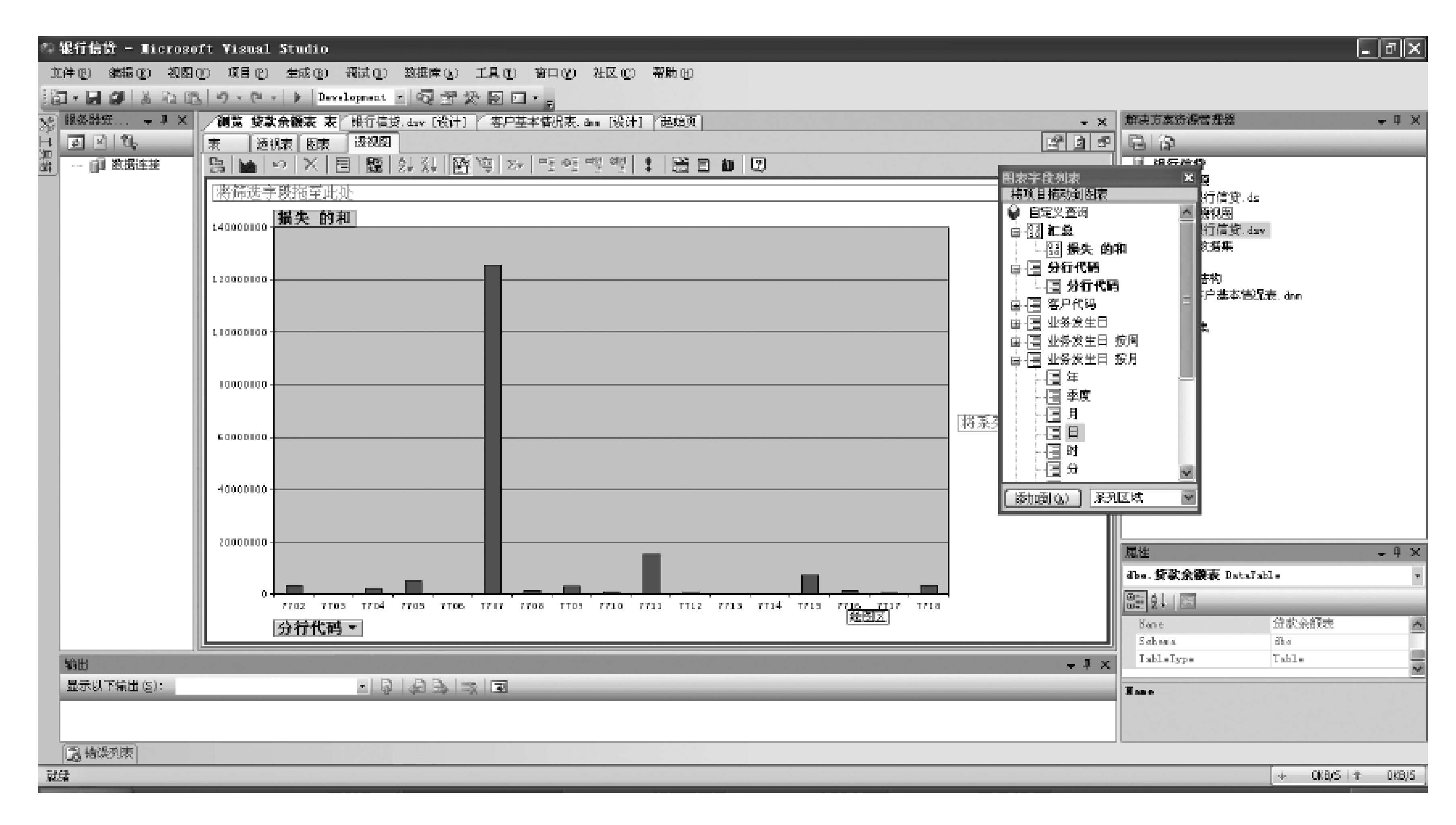


图 3-35　分行代码和损失

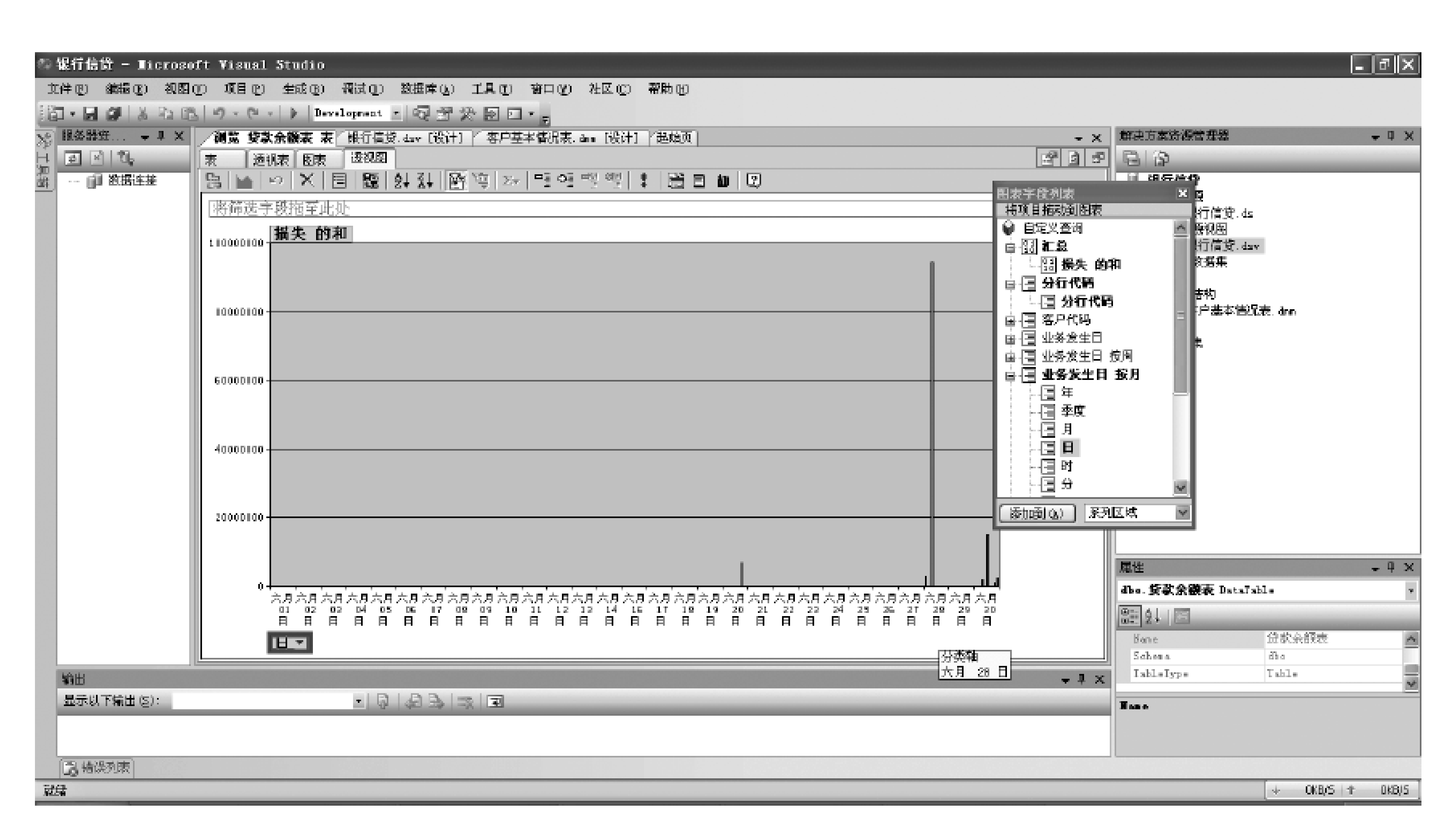


图 3-36　三维分析(1)

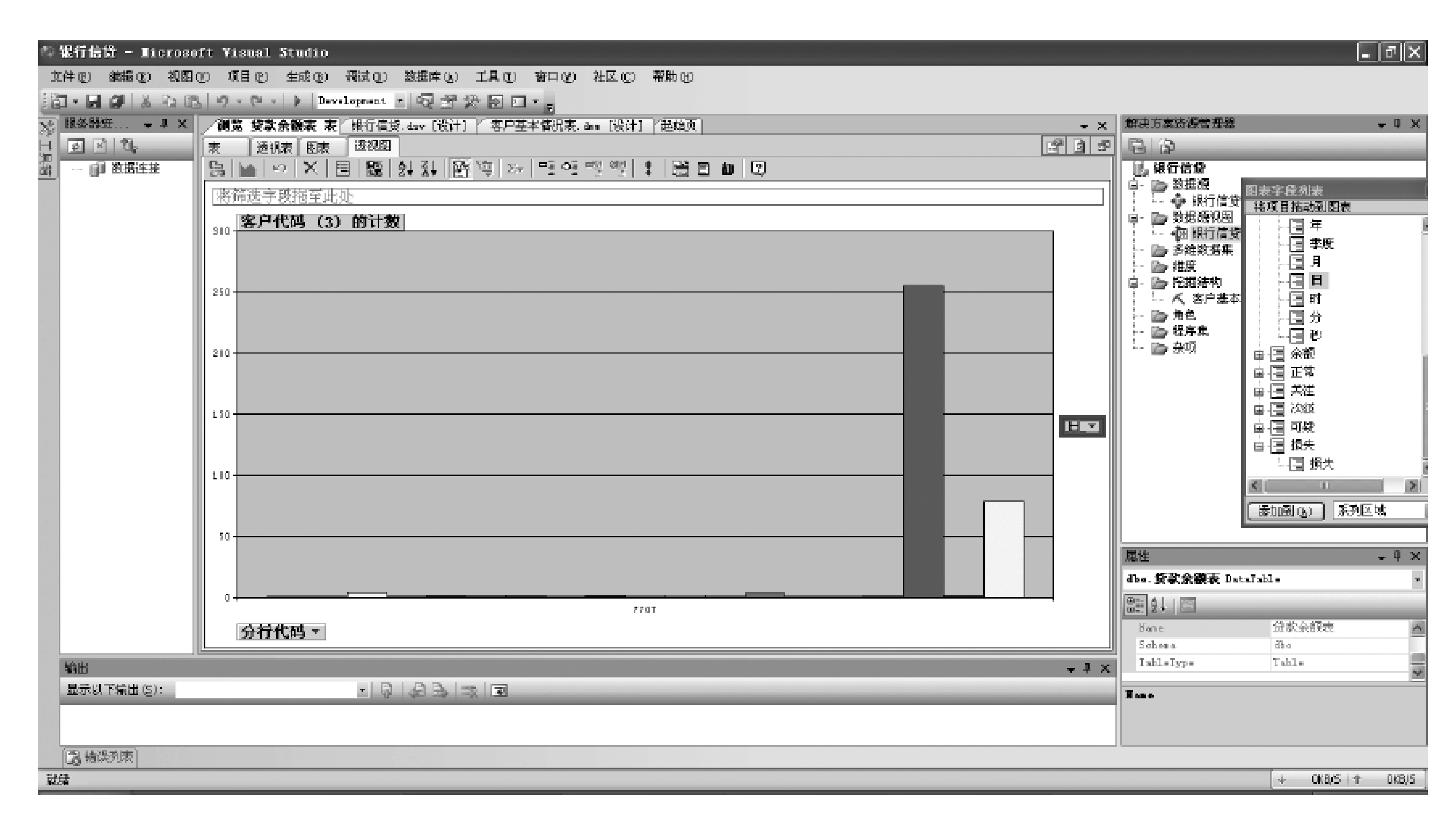


图 3-37　三维分析(2)

分析可得代码为7707的分行，在6月28日的客户代码计数为255，所以可以把这255家企业的情况分析作为工作的重点。

小结

本章首先介绍了联机分析处理技术的定义、特征和维、数据单元等一些基本概念。然后介绍了联机分析处理中的钻取、切片切块、旋转等多维分析操作。接下来又介绍了联机分析处理的基本数据类型，包括多维联机分析处理、关系联机分析处理和混合型联机分析处理。最后具体介绍了基于SQL Server 2005的联机分析处理实现，使读者了解了联机分析处理的应用过程。

习题3

1. 什么是联机分析处理？
2. 多维数据结构一般包括哪些内容？常用的多数据分析方法有哪些？
3. 简述OLAP的评价准则。
4. OLAP的基本多维分析操作有哪几种？
5. 请比较ROLAP与MOLAP在数据存储、技术及特点上的不同。

第 4 章　数据预处理

数据挖掘是随着数据库技术和人工智能技术的进步而发展起来的一门新兴学科，其处理对象是大量的日常业务数据。原始业务数据是数据挖掘的信息来源，而这些数据通常含有噪声、大量的空缺值和不一致现象，影响数据挖掘的效率和结果的有用性，甚至产生一些无效归纳。对原始数据进行预处理，为数据挖掘过程提供干净、准确、简洁的数据，提高挖掘效率和准确性，是数据挖掘中非常重要的环节。

4.1　数据预处理概述

大型现实世界数据库的一个共同特点是存在大量不完整的、含噪声的和不一致的数据。用于数据挖掘的原始数据源可能是多个数据库或数据仓库，而这些数据源的结构和规则可能是不同的，这将导致原始数据非常的杂乱、不可用；即使在同一个数据库中，也可能存在重复的和不完整的数据信息。为了使这些数据能够符合数据挖掘的要求，从而提高效率并得到清晰的结果，必须进行数据预处理。

4.1.1　原始数据中存在的问题

归结起来，原始数据主要存在以下几个方面的问题。

1. 不一致

由于原始数据来源于多个不同的应用系统或数据库，信息庞杂，采集和加工的方法有别，数据描述的格式也各不相同，缺乏统一的分类标准和信息的编码方案，难以实现信息的集成共享，很难直接用于数据挖掘。

2. 重复

同一事物在数据库中存在两条或多条完全相同的记录，或者相同的信息冗余地存在于多个数据源中。

3. 不完整

实际应用系统中，由于系统设计的不合理或者使用过程中的某些因素，某些属性值可能会缺失或者不确定，比如，认为某些属性不重要而在数据输入时忽略了，或者没有能够获得确定的值而空缺。

4. 含噪声

噪声是指一个测量变量中的随机错误或偏离期望的孤立点值，产生噪声的原因很多，比如，人为的、设备的和技术的等。

5. 维度高

原始数据中通常记录事物较为全面的属性，而在一次数据挖掘中，这些属性并不全是有用的，只需要一部分属性就可以得到期望知道的知识，而且，无用属性的增加还可能会导致无效归纳，把挖掘结果引向错误的结论。

6. 数据不平衡

某类样本数量明显少于其他类样本数量的数据集称为不平衡数据集。不平衡数据集的分类问题大量存在于人们的现实生活和工业生产之中，如网络入侵、医疗诊断、信用卡欺诈检测、语音处理、信息检索和文本分类等。

因此，直接把原始数据用于数据挖掘是不实际的，一个完整的数据挖掘系统应该提供数据预处理模块，此模块的功能是为数据挖掘的算法提供完整、干净、准确、更有针对性的数据，减少算法的计算量，提高挖掘效率和准确程度，即形成供数据挖掘算法使用的目标数据知识基。知识基仍然以二维表的形式存在，由若干筛选出来的元组和属性组成，包含了与挖掘任务相关的所有数据的特征。知识基形成的前提是根据用户的需要，确定挖掘任务，在领域专家的建议和指导下，采用合适的方法重新组织原始数据，使之能够最大程度上支持数据挖掘算法。

4.1.2 数据预处理的方法和功能

数据预处理包含数据清洗、数据集成、数据变换和数据归约几种方法，本节简要介绍各种方法的概念，具体的内容和例子将在下面各节中给出。

1. 数据清洗(data cleaning)

现实世界中的数据通常是“脏的”，通过数据清洗过程填充空缺值，识别孤立点，去掉原始数据中的噪声和无关数据。

例如，在客户数据中，一般都包括客户的性别属性，如“男”或“女”，但也经常包含诸如“未知”之类的值或者空值，这些噪声数据就需要通过数据清洗来保证数据质量。

2. 数据集成(data integration)

数据集成是将多个数据源中的数据结合起来存放在一个一致的数据存储中，数据集成涉及多个数据源的数据匹配问题，数值冲突问题和数据的冗余问题等。

例如，在客户数据中，一个数据源可能用千元表示客户收入，而另一个数据源可能使用元表示同一个属性。

3. 数据变换(data transformation)

数据变换是把原始数据转换成为适合数据挖掘的形式。包括对数据的汇总和聚集、概化、规范化，还可能需要进行属性的构造。

4. 数据归约(data reduction)

数据归约技术用于产生数据的归约表示，使得数据量减小，更适合于数据挖掘算法的需要，并且能够得到和原始数据相同的分析结果。用到的主要方法包括数据立方体聚集、维归约、数据压缩、数值归约、离散化和概念分层等。

4.2 数据清洗

4.2.1 属性选择与处理

数据挖掘中使用的数据不必是所有的原始数据，特别是当一些属性明显和挖掘目的无关时，使用整个原始数据反而会降低挖掘效率，甚至产生无效规律，所以，应该从原始数据中

选取合适的属性进行数据挖掘。选取过程通常根据行业知识或专家意见进行，一般应依据以下几个原则。

1. 尽可能赋予属性名和属性值明确的含义

通常，在现实数据库中，有些属性名称和属性值的含义不是很明确，只能够被操作人员记住和理解；而实施数据预处理和数据挖掘的人一般不可能是操作人员，在数据预处理的初期，首先要对名称和取值含义含糊的属性进行处理，赋给它们具有明确含义的名称和取值，便于理解和使用。

2. 统一多数据源的属性值编码

在一次数据挖掘过程中，可能会涉及多个数据源的多张表，所以要保证在各个数据源中对同一事物特征的描述是统一的。

例如，在多个数据源中描述性别的属性，一个数据源用“男”、“女”作为该属性的值，另一个数据源可能使用“0”、“1”来表示，第三个数据源可能使用“M”、“F”作为属性值，在多个数据源合并的时候，就需要把这些属性值统一起来。

3. 处理唯一属性

一般来说，原始数据中的关键属性或唯一属性对数据挖掘是无用的，它们通常用来作为记录的唯一性标识，不形成规则，可以去除。但是，如果需要建立挖掘结果和原始数据之间的直接对应关系的话，通常要保留一个或多个必需的关键属性或唯一属性。

4. 去除重复属性

有时候，原始数据中会出现意义相同或者可以用于表示同一信息的多个属性，如年龄和出生日期。在一次数据挖掘中，如果只考察某些客户的年龄段和消费特征的关系，这两个属性就是重复的，只要选取一个就可以满足需要。当然，在某次挖掘中可能同时需要这些重复属性，如考察客户年龄段和出生月份(或季节)对消费特征的影响，此时这两个属性就表示不同的信息，应该同时保留。

5. 去除可忽略字段

当一个属性值缺失非常严重，只有极少数值保存下来时，该属性已经不能形成任何有用的知识，但是数据挖掘算法反而会认为这些大量的空值形成了有用的知识。所以这样的属性应该去除。

6. 合理选择关联字段

如果属性 X 可以由另一个或多个属性推导或者计算出来，则认为这些字段之间的关联度高，属性 X 和它的关联属性对数据挖掘的作用是相同的，所以只选择其中之一，或者属性 X，或者它的关联属性。

如当职工的月薪和有薪月份固定时，月薪、有薪月份和年薪之间就形成了高度的关联，此时应只保留月薪、有薪月份或者年薪；另如，商品的价格、数量和总价格，也形成高度关联关系。

经过以上处理之后，还需要对已经选择的属性进一步处理，去掉数据中的噪声、填充空值、丢失值和处理不一致数据。

4.2.2 空缺值处理

如果一些有用的属性因某种原因，没有记录值，那么必须在数据清洗中对这些空缺值进

行处理，处理的方法有下面几种。

1. 忽略该记录

当一个记录中有多个属性值空缺、特别是关键信息丢失时，即使采用某种方法把所有缺失的属性值填充好，该记录也已经不能反映真实的情况，对于数据挖掘算法来说，这样的数据性质是很差的，应该忽略该记录。

2. 去掉属性

如果所有记录中的某一个属性值缺失严重，可以认为该属性对于知识发现来说已经没有意义，此时，一个有效的清洗方法就是把该属性排除在挖掘数据集之外。

3. 写空缺值

以某些背景资料为依据，手工地填写空缺值。这种方法的优点是能够得到真实的数据，但是耗费人力很大，而且速度也慢，不能用来处理较大的和值缺失较多的数据集。

4. 使用默认值

对于离散值属性，用一个常数取代空缺值，表示这个属性值是未知的，如 unknown。这种方法的优点是简单、易实现，但是如果对于空缺较多的属性，所有空缺都用这个默认值代替，挖掘算法很可能认为形成了一个有用的知识，导致挖掘得出无用的规律，所以应尽量少用此方法。

5. 使用属性平均值

对于连续属性，计算所有记录的该属性平均值，用来填充空缺值。

6. 使用同类样本平均值

计算同类样本记录的该属性平均值，用来填充空缺值。

7. 预测最可能的值

此方法是一种最常用的方法，它从现有数据的多个信息推测空缺值。根据其他完整的记录数据，使用一定的预测方法，得到最可能的预测值。

一些数据挖掘算法在处理空值方面的能力比较强，如决策树算法、关联规则算法等，能够快速地产生较为准确的知识模型，而其他算法则可能花费较长的时间，而且产生的模型精确性差一些，如神经网络方法。另外，由于数据库管理系统之间存在一些差异，不同的数据库系统对空值的处理可能不同，比如 Oracle 数据库不区分空值和空字符串。在进行数据预处理时，要考虑挖掘算法和数据库系统的特点，选择合适的预处理方法。

4.2.3 噪声数据处理

在测量一个变量时可能产生一些误差或者错误，使得测量值相对于真实值有一定的偏差，这种偏差称之为噪声。为了去除这些噪声，使数据接近真实值，可以采用下面的一些方法。

1. 分箱(binning)

分箱方法是一种简单常用的预处理方法，通过考察相邻数据来确定最终值。把待处理的数据(某列属性值)按照一定的规则放进一些箱子中，考察每一个箱子中的数据，采用某种方法分别对各个箱子中的数据进行处理。

所谓“箱子”，实际上就是按照属性值划分的子区间，如果一个属性值处于某个子区间范围内，就称把该属性值放进这个子区间代表的“箱子”内。在采用分箱技术时，需要确定的两

个主要问题是，如何分箱以及如何对每个箱子中的数据进行平滑处理。分箱之前需要对记录集按目标属性值的大小进行排序。下面首先介绍分箱的方法。

（1）统一权重。也称等深分箱法，将数据集按记录行数分箱，每箱具有相同的记录数，每箱记录数称为箱的权重，也称箱子的深度。这是最简单的一种分箱方法。

（2）统一区间。也称等宽分箱法，使数据集在整个属性值的区间上平均分布，即每个箱的区间范围是一个常量，称为箱子宽度。

（3）最小熵。使在各区间分组内的记录具有最小的熵。熵是信息理论中数据无序程度的度量标准，提出信息熵的基本目的，是找出某种符号系统的信息量和冗余度之间的关系，以便能用最小的成本和消耗来实现最高效率的数据储存、管理和传递。

某个字符（或数值）的信息量的基本计算公式如下：

$$I = -\operatorname{lb}(P) \tag{4-1}$$

其中，I 表示信息量，P 表示某种字符出现的概率，信息量的单位是比特（bit，二进制的 0 和 1）。数据集的熵用下面的公式计算：

$$H = \sum_i p_i \operatorname{lb}(1/p_i) \tag{4-2}$$

数据集的熵越低，说明数据之间的差异越小，最小熵划分就是为了使每箱中的数据具有最好的相似性。

给定箱的个数，如果考虑所有可能的分箱情况，最小熵方法得到的箱应该是具有最小熵的分箱。

（4）用户自定义区间。当用户明确希望观察某些区间范围内的数据分布时，可以根据需要自定义区间，使用这种方法可以方便地帮助用户达到目的。

考虑下面的例子：

在选定的数据集中，客户收入属性 income 排序后的值（人民币元）：800 1000 1200 1500 1500 1800 2000 2300 2500 2800 3000 3500 4000 4500 4800 5000，对此记录集分别应用上述分箱技术，观察分箱后的结果。

统一权重：设定权重（箱子深度）为 4，分箱后

箱 1：800　1000　1200　1500

箱 2：1500　1800　2000　2300

箱 3：2500　2800　3000　3500

箱 4：4000　4500　4800　5000

统一区间：首先确定箱子的数目，比如 4，根据数据集的取值范围[800，5000]，每个箱子的宽度为（5000－800）/4，得到 4 个宽度相等的子区间：[800，1850）、[1850，2900）、[2900，3950）和[3950，5000）。分箱后

箱 1：800　1000　1200　1500　1500　1800

箱 2：2000　2300　2500　2800

箱 3：3000　3500

箱 4：4000　4500　4800　5000

用户自定义：如将客户收入划分为 1000 元以下、1000～2000、2000～3000、3000～4000 和 4000 元以上几组，分箱后

箱 1：800

箱 2：1000　1200　1500　1500　1800　2000

箱 3：2300　2500　2800　3000

箱 4：3500　4000

箱 5：4500　4800　5000

实际应用中，要根据考察目的选用合适的分箱方法。

分箱目的是对各个箱子中的数据进行处理，所以完成分箱之后，就要考虑选择一种方法对数据进行平滑，使得数据尽可能接近。通常使用 3 种方法进行数据平滑：按平均值平滑、按边界值平滑和按中值平滑。将上例用统一区间方法分箱后的结果，分别采用 3 种平滑方法进行处理。

（1）按平均值平滑。对同一箱值中的数据求平均值，然后用这个平均值替代该箱子中的所有数据。应用此平滑方法，平滑后的结果如下：

箱 1：1300　1300　1300　1300　1300　1300

箱 2：2400　2400　2400　2400

箱 3：3250　3250

箱 4：4575　4575　4575　4575

（2）按边界值平滑。对于箱子中的每一个数据，观察它和箱子两个边界值的距离，用距离较小的那个边界值替代该数据。用此方法平滑后的结果如下：

箱 1：800　800　800　1800　1800　1800

箱 2：2000　2000　2800　2800

箱 3：3000　3500

箱 4：4000　4000　5000　5000

当某个数据与左右两个边界值的距离相等时，可以约定用其中一个边界替代。本例中使用左边界，如箱 4 中的第 2 个数据 4500 用左边界 4000 代替。

（3）按中值平滑。取箱子的中值，用来替代箱子中的所有数据。中值也称中数，将一些数据排序之后，如果这些数据是奇数个，中值就是位于最中间位置的那一个；如果是偶数个，中值应该是中间两个数的平均值。用此方法平滑后的结果如下：

箱 1：1350　1350　1350　1350　1350　1350

箱 2：2400　2400　2400　2400

箱 3：3250　3250

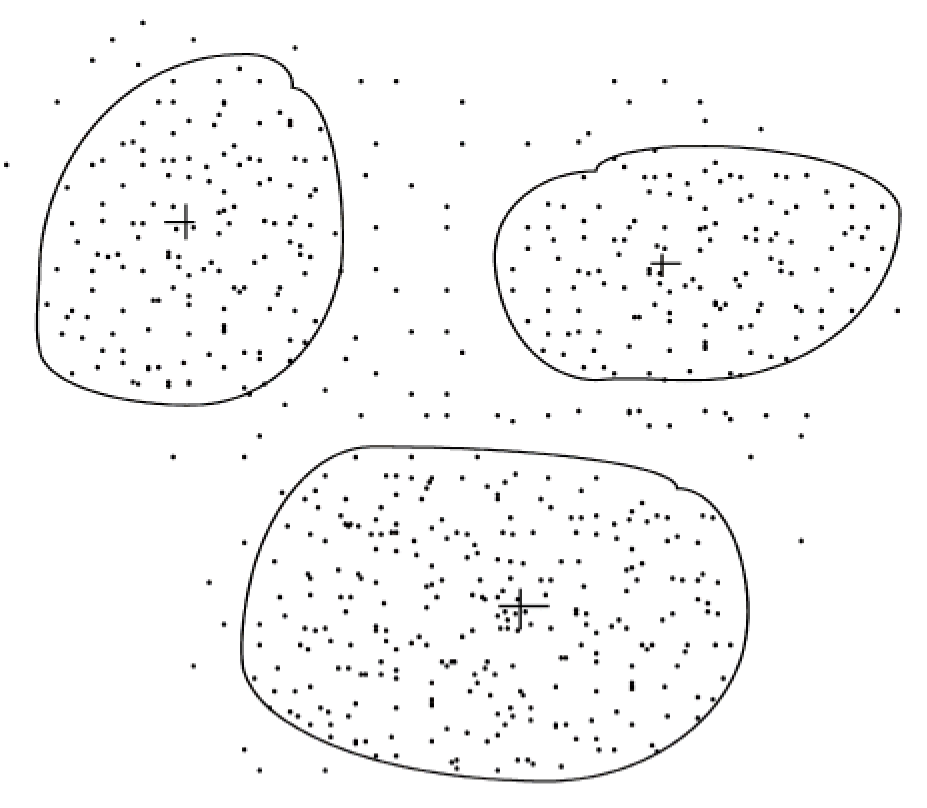
箱 4：4650　4650　4650　4650

2. 聚类（clustering）

去除噪声数据的第二个方法是使用聚类，聚类是将物理的或抽象对象的集合分组为由类似的对象组成的多个类的过程，聚类的结果是生成一组由数据对象组成的集合，称为簇。同一个簇中的所有对象具有相似性，并且一个对象与同簇中任何一个对象之间的相似性一定强于它与其他簇中任何一个对象之间的相似性。用聚类方法去除噪声，就是要找出那些落在簇之外的值，称为孤立点，这些孤立点被视为噪声。而对同一簇中的对象用相同的特征来标识。

聚类方法不需要任何先验知识，直接形成簇并对簇进行描述，关于聚类算法，本书的后

续章节将详细讲解。图 4-1 是聚类的一个例子。


图 4-1 用聚类方法去掉噪声

图中共形成了 3 个聚类,“+”号用来表示聚类的质心。聚类的质心就是聚类中的平均点。不在任何聚类中的点称为孤立点,就是要去掉的噪声数据。

3. 回归(regression)

回归试图发现相关的变量之间的变化模式,通过使数据适合一个函数来平滑数据,即通过建立数学模型来预测下一个数值,回归方法分为线性回归(linear regression)和非线性回归(nonlinear regression)。线性回归又称简单回归,是最简单的回归形式,将一个变量看作另一个变量的线性函数,如 $Y=aX+b$,其中,Y 称为因变量,X 称为自变量,a,b 称为回归系数,可用最小二乘法求得,使得实际数据与模型之间误差为最小。对于随机的时序数据可用分段线性逼近的方法划分置信区,使得该区间中的直线模型同实际数据的误差在规定范围内。只要用户指定精度要求,就可以对时序数据建模,得到一系列的分段线性模型,它们由回归系数 a,b 及分段时间戳确定。

多元回归是线性回归的扩展,也称复回归,有两个或两个以上自变量。当预测涉及多个属性字段时,就应该考虑使用多元回归,如 $Z=aX+bY+c$,回归系数同样也可以使用最小二乘法求解。

有关回归方法的详细介绍参见本书第 7.4 节。

4.2.4 不平衡数据的处理

不平衡数据分类考虑的是各类样本数目不平衡情况下的分类学习问题。以二分类为例,若其中有一类(正类、多数类)的学习样本比另一类(负类、少数类)的学习样本多得多,那么就称这样的分类问题为不平衡分类问题。不平衡数据在实际应用中经常碰到,如欺诈识别、入侵检测、医疗诊断以及文本分类等都是典型的不平衡数据问题。传统的分类方法主要考虑的是各类学习样本数量大致均衡的情形,其评价标准主要是基于精度的。这使得现有的分类方法往往不能有效地处理不平衡数据,尤其是数据存在严重不平衡时(正/负类学习样本数量比可高达 100∶1、1000∶ 1 甚至 10000∶1)。以下是两种用于处理不平衡数据的抽样技术,基本思想是通过改变训练数据的分布来消除或减小数据的不平衡。

(1) 过抽样(oversampling)。处理不平衡数据的最常用方法,过抽样方法通过增加少数类样本来提高少数类的分类性能,最简单的办法是复制少数类样本,缺点是引入了额外的训练数据,会延长构建分类器所需要的时间,没有给少数类增加任何新的信息,而且可能会导致过度拟合。

(2) 欠抽样(undersampling)。欠抽样方法通过减少多数类样本来提高少数类的分类性能,最简单的方法是通过随机地去掉一些多数类样本来减小多数类的规模,缺点是会丢失多数类的一些重要信息,不能够充分利用已有的信息。

4.3 数据集成和变换

数据挖掘所使用的数据通常来自于多个数据存储,所以经常需要把多个数据存储合并起来,这个过程称为数据集成。而为了使数据符合算法和挖掘目标的需要,如数据的取值范围、粒度等,还需要对数据进行变换。本节讲述数据集成和数据变换的相关内容。

4.3.1 数据集成

数据集成是将多文件或者多数据库中的异构数据进行合并,然后存放在一个一致的数据存储中,主要工作涉及数据的冲突问题和不一致数据的处理问题。

用于数据挖掘的数据可能来自于多个实际应用系统,这些应用系统可能是异构的,还可能存在属性的同名不同义、同义不同名、单位不统一、类型不一致等问题,在数据集成的过程中,需要发现和统一这些矛盾,对原始数据进行重新组织,形成挖掘数据。

在数据集成过程中,通常需要考虑以下几个问题。

1. 模式匹配

通过下面的例子来说明模式匹配问题。

表 4-1 和表 4-2 是两张原始数据表,分别存储“客户基本情况”和“客户交易数据”,其中“客户基本情况”表中包含客户标志、客户收入水平、客户类型等,而“客户交易数据”表中包含客户标志和客户交易的细节,如时间、商品类型、金额等。

表 4-1 客户基本情况表

属性名称	数据类型	说 明
id	Short int	客户标志
gender	boolean	性别
birth	data	出生日期
type	boolean	是会员
income	Short int	月收入(元)

表 4-2 客户交易数据表

属性名称	数据类型	说 明
customer_id	int	客户标志
time	date	交易日期
goods	string	商品名称
price	real	商品价格
count	short int	商品数量
total_price	real	总价格

用户希望发现客户背景和客户购买类型、购买力的关系,针对数据挖掘的需要,数据预处理时,需要把这两张表集成为一个挖掘数据源。

从属性说明中可以看出,两张表中的数据可以通过“客户标志”关联起来,但是如表中所示,这两张表中的客户标志属性名称不同,所以在集成时必须通过可靠的手段确定 id 和 customer_id 是同一个信息,通常可以通过和用户讨论的方法来确定,因为用户是数据的使用者,他们应该理解属性的含义。如果数据库可以提供元数据,则元数据应该是确定属性含义的最好依据。例如,将表 4-1 中的 id 字段名改为 customer_id。

另外,属性“客户基本情况. id”和“客户交易数据. customer_id”的类型也不相同,必须统一为相同的数据类型,在保证不丢失信息的基础上,应该选择长度较小的数据类型,当数据集很大时,采用较短的数据类型可以节省系统的开销。

模式匹配问题还体现在自动采集数据和人工录入数据的差别上,特别是和时间相关的

数据上，自动采集数据的时间戳较密，而人工录入数据的时间戳较稀；事务处理的数据时间是非等间隔的，而数据恢复的时间戳是历史的。这些不同的模式如何匹配必须要有元数据加以说明，才能避免数据集成时带来的模式匹配错误。

2. 数据冗余

冗余是指重复存在的信息，数据冗余的存在使得挖掘程序需要对相同的信息进行重复的处理，增加了数据挖掘的复杂性，降低挖掘效率。最明显的数据冗余是挖掘数据中存在两个或多个重复的记录，这种冗余也称为重复。

在一次挖掘中，有多个属性同时对挖掘结果产生影响，而对同一个结果有影响的属性，它们之间比较容易产生关联。那些可以由其他属性推导得出的属性，被认为是冗余属性。

如上面"客户交易数据"表中的 total_price 属性，它实际上可以由商品价格和数量两个属性计算得到。如果挖掘目的是为了使用客户消费金额信息，total_price 属性对挖掘结果的贡献等同于 price 属性和 count 属性综合作用的贡献，此时就产生了数据冗余。可以舍弃 total_price 属性，保留 price 属性和 count 属性，或者舍弃 price 属性和 count 属性，而保留 total_price 属性。这是比较明显的数据冗余现象，通常在理解属性含义的基础上，稍加观察就可以发现，有时这种冗余也发生在不同的表之间。

有些数据的冗余比较隐蔽，不容易直观的观察出来。如果希望通过发现属性之间的关联程度来确定是否发生了冗余现象，可以使用相关分析方法。相关分析方法检测一个属性蕴涵另一个属性的可能性，这种可能性越大，表明属性之间的蕴涵关系越明显，可以去掉其中的一个属性而只保留另一个属性。例如，属性 X 和 Y 的相关性可以由下式来度量：

$$r_{X,Y} = \frac{\sum (X-\bar{X})(Y-\bar{Y})}{(n-1)\sigma_X \sigma_Y} \tag{4-3}$$

其中，n 是记录个数，$\bar{X}$ 和 $\bar{Y}$ 分别是 X、Y 的平均值，σ_X 和 σ_Y 是 X 和 Y 的标准差。

$$\bar{X} = \frac{\sum X}{n}, \quad \bar{Y} = \frac{\sum Y}{n} \tag{4-4}$$

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\bar{X})^2}{n-1}}, \quad \sigma_Y = \sqrt{\frac{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}{n-1}} \tag{4-5}$$

3. 数据值冲突

在多个数据源中，表示同一实体的属性值可能不同，这些不同表现在数据值、数据类型、数量单位或编码等方面，例如对于客户的收入，在一个数据源中可能表示为元，而在另一个数据源中可能表示为千元；另外，对于客户类型（是否会员客户），一个数据源可能用布尔类型数据表示，而另一数据源中可能用字符类型表示，即使是用同一种数据类型，由于不同的数据库管理系统对数据类型的处理不同，也可能发生数据值冲突。

4.3.2 数据变换

通常，原始数据表中的数据不适合直接用于数据挖掘，需要对它们进行变换之后才能使用，数据变换涉及多个方面。下面详细介绍数据变换的平滑、聚集、数据概化、规范化和属性构造等主要内容。

1. 平滑(smoothing)

平滑即去除噪声,还可以将连续的数据离散化,增加粒度。如采用分箱或聚类方法时,实际上是把一个区域内的值用同一个数值表示,在一定的误差允许条件下减少了属性的取值个数,进而减少挖掘算法的工作量。数据平滑的方法包括分箱、聚类、回归等,具体方法已经在第 4.2.3 小节介绍过。

2. 聚集(clustering)

聚集即对数据进行汇总,例如,当分析客户的背景情况对购买能力的影响时,只需要关心客户消费的金额,并不需要了解客户购买了什么商品以及商品的数量、价格等信息,对于上面给出的"客户交易数据"表,需要汇总客户每次交易的货物总价,进而汇总客户所有交易的总金额。聚集常常用来构造数据立方体。

3. 数据概化(generalization)

通常,从原始数据集得到的数据包含一些低层概念的描述,而在数据挖掘中有时并不需要细化到这些概念,其存在会使数据挖掘过程花费更多的时间,增加复杂度,可以用它的高层概念替换,也就是数据概化。

例如,客户的出生日期通常是保存在客户背景数据存储中的,为了了解不同年龄段客户的消费特点,只对客户的出生年代(或者年龄阶段)感兴趣,并不需要知道客户的具体出生日期,所以"出生日期"可以概化为"出生年代"。

4. 规范化(normalization)

将数据按比例缩放,使之落入一个特定的区域,如 0.0～1.0,称为规范化,或者标准化。规范化对基于距离的聚类算法和神经网络算法是非常重要的,可以保证输入值在一个相对小的范围内,加快训练速度;另外不会发生因为输入值的范围过大而使权重过大的情况。

例如,在应用聚类方法时,数据的度量单位将对聚类的结果产生很大的影响,数据的度量单位越小,数值的取值就越大,对聚类产生的影响也越大。为了避免这种情况的发生,一个办法就是对数据进行规范化,把数据规范化到一个无单位的特定区域中。当采用神经网络方法时,需要事先告知变量的变化范围(如使用某些数据挖掘工具时,要求提供变量范围),神经网络在这个变化范围内跟踪数据的变化,学习规律。而在预测时,一旦有超出该范围的数据,神经网络就不能准确地跟踪。有时,还会放大数据的范围以增强对比效果。

下面讨论几种常用的数据规范化方法。

(1) 最小—最大规范化(MIN—MAX normalization)。假设数据的取值区间为[old_min,old_max],最小最大规范化即把这个区间映射到新的取值区间[new_min,new_max]。对于任意一个在原来区间中的变量,在新的区间中都有一个值和它对应,这是一个线性变换过程,变量被映射到新区间的值通过下面的公式计算得出:

$$x' = \frac{x - \text{old_min}}{\text{old_max} - \text{old_min}}(\text{new_max} - \text{new_min}) + \text{new_min} \tag{4-6}$$

其中,x 是属性的真实值,x'是规范化后的值。

例如,"客户背景数据"表中的客户月收入 income 属性的实际值范围为[430,8000],需要把这个属性值规范到[0, 1],对属性值 3200 应用上述公式:

$$x' = \frac{3200-430}{8000-430}(1.0-0)+0 = 0.365918$$

根据精度要求保留小数(假设精度要求 0.0001),最终取值 0.3659 就是属性值 3200 规范后的值。

应用最小-最大规范化的前提条件是属性的取值范围必须已知,如果取值超出了给定的范围,其产生的规范值将超出约定的区间范围,发生越界错误。

(2) 零-均值规范化(z-score normalization)。零-均值规范化即根据属性值的平均值和标准差进行规范化,即

$$x' = \frac{x-\overline{X}}{\sigma_X} \tag{4-7}$$

其中,$\overline{X}$ 为所有样本属性值的平均值,而 σ_X 为样本的标准差。当属性值范围未知的时候,可以使用此方法进行规范化。

例如,第 4.2.3 小节中的客户收入数据:800 1000 1200 1500 1500 1800 2000 2300 2500 2800 3000 3500 4000 4500 4800 5000,采用零-均值规范化方法进行规范化:

首先,求样本的平均值 $\overline{X}$:

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n} = 2637.5$$

样本的标准差 σ_X

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\overline{X})^2}{n-1}} = 1333.17$$

如果对属性值 3000 进行零-均值规范化,可得到

$$x' = \frac{x-\overline{X}}{\sigma_X} = \frac{3000-2637.5}{1333.17} = 0.2719$$

(3) 小数定标规范化(decimal scaling normalization)。通过移动属性值的小数点位置进行规范化。此方法也需要在属性取值范围已知的条件下使用,小数点移动的位数根据属性的最大绝对值确定,具体计算公式为

$$x' = \frac{x}{10^{\alpha}} \tag{4-8}$$

其中,α 是使 $\mathrm{Max}(|x'|)<1$ 的最小整数。

仍以第 4.2.3 小节中的客户收入数据为例,该样本数据值的范围为 800～5000,最大绝对值为 5000,为了使得

$$\max\left(\left|\frac{x}{10^{\alpha}}\right|\right) = \max\left(\left|\frac{5000}{10^{\alpha}}\right|\right) < 1$$

成立,α 取 4,规范化后最大值 5000 的值为 0.5。

5. 属性构造

为了提高数据挖掘的精度或者使数据结构更容易理解,有时候会根据已有的属性构造新的属性添加到挖掘数据集中,例如在客户背景数据表中,根据客户月收入,构造“收入水平”属性,取值为低、中、高。这种方法对分类算法有帮助。

4.4 数据归约

数据归约用于从源数据集中得到数据集的归约表示，从原始数据选择出来的数据集非常大，甚至使得在其上进行数据挖掘非常困难，数据归约的目的是为了减少原始数据量，在不破坏数据完整性的前提下，获得比原始数据小得多的挖掘数据集，该数据集可以得到与原始数据相同的挖掘结果。

4.4.1 数据归约的方法

有多种方法用于数据归约，本节着重介绍以下几种方法。

(1) 数据立方体聚集。把聚集的方法用于数据立方体。

(2) 维归约。检测并删除不相关、弱相关或冗余属性。

(3) 数据压缩。选择正确的编码压缩数据集。

(4) 数值压缩。用较小的数值表示数据，或采用较短的数据单位，或者用数据模型代表数据。

(5) 离散化和概念分层生成。使连续的数据离散化(discretization)，用确定的有限个区段值代替原始值；概念分层是指用较高层次的概念替换低层次的概念，以此来减少取值个数。

4.4.2 数据立方体聚集

数据立方体是数据的多维建模和表示，由维和事实组成。维就是涉及的属性，而事实是一个具体的数据。通常认为立方体是一个 3 维的几何结构，实际上，一个数据立方体的维数可以是 n 维。表 4-3 中的数据表“销售记录”中，记录了各种商品在全国各个省份的销售情况。

表 4-3 “销售记录”表结构

属性名称	数据类型	长度	说　明
goods_type	string	10	商品类型
year	string	4	年份
province	string	20	省份
sales	real	6	销售金额(万元)

以表中的商品类型、年份和省份作为立方体的维，销售金额为事实构造的 3 维数据立方体如图 4-2 所示。

如果挖掘时感兴趣的信息是年度总销售量，不关心每个省份的销售量，就可以对上面构造的立方体进行聚集，得到一个 2 维数据立方体，如图 4-3 所示。这个 2 维立方体表示了各类商品在各个年度中的销售金额。

如果需要，还可以把 2 维数据立方体进一步聚集成 1 维。

4.4.3 维归约

在数据立方体的概念基础上，把属性称为维，维归约即去掉不相关的属性，减少数据挖

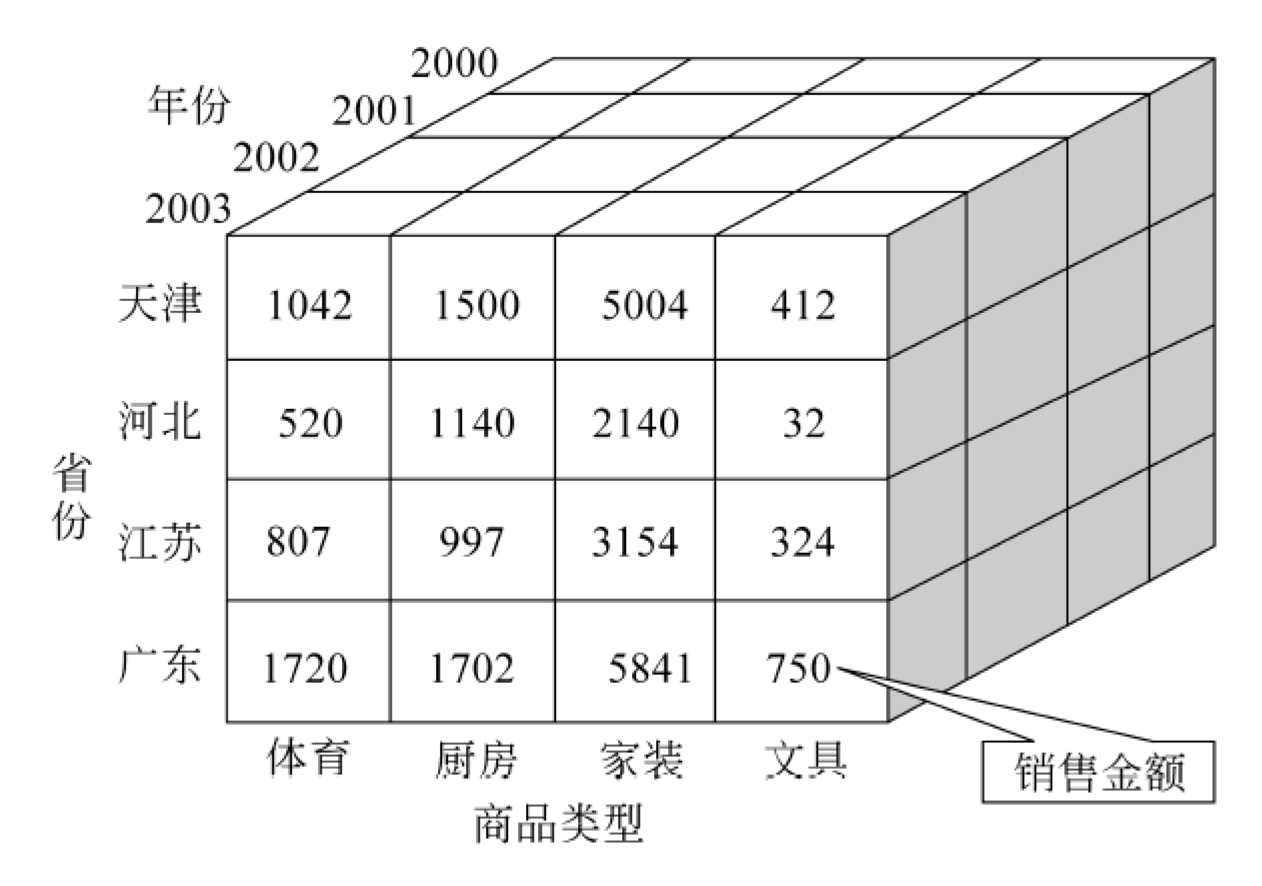
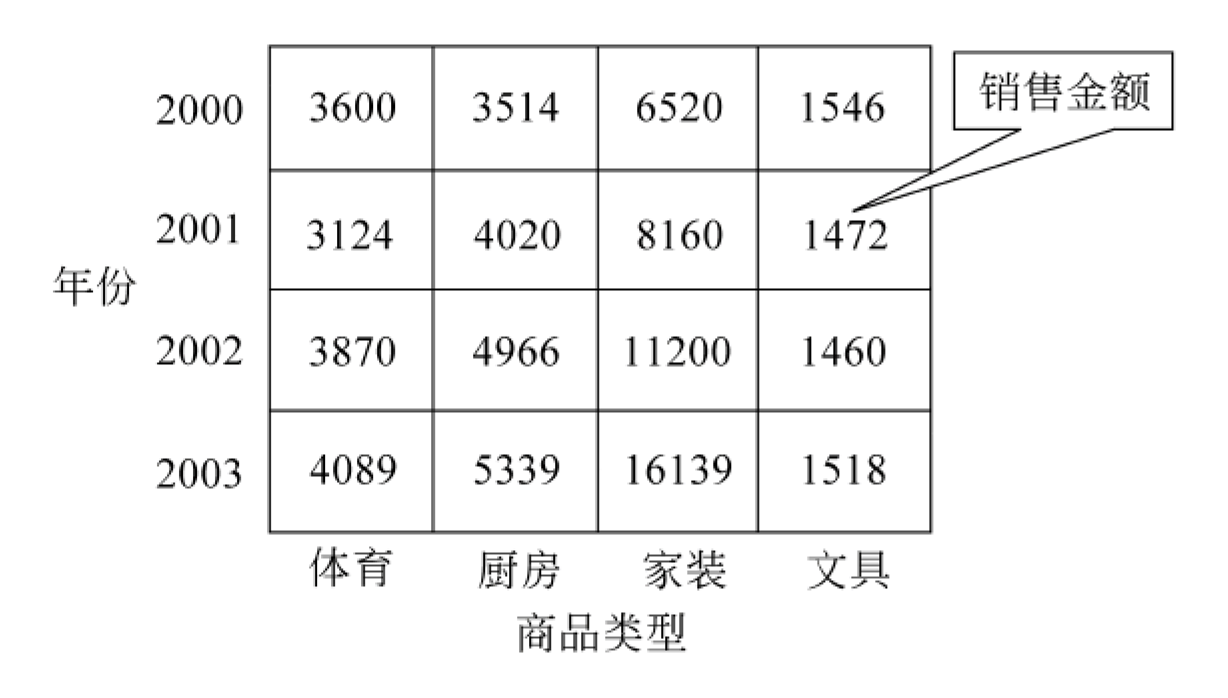


图 4-2　销售数据立方体



年份	体育	厨房	家装	文具
2000	3600	3514	6520	1546
2001	3124	4020	8160	1472
2002	3870	4966	11200	1460
2003	4089	5339	16139	1518

图 4-3　聚集后的销售数据立方体

掘处理的数据量。

为什么要进行维归约？因为数据中含有数十个属性，而在一次挖掘中，只有其中小部分属性与挖掘任务相关，前面也曾经叙述过，无关的数据会减慢数据挖掘的速度，甚至产生无用的规律，影响结果分析，所以要去掉这些属性，即维归约。

那么，如何进行维归约呢？一个直接的办法就是人工选择属性，但是如果涉及很多属性的话，人工方法效率低而且费时。实际上去除无关属性可以通过选择相关属性来完成，找到一个最小属性子集，使得这个子集能够具有和原数据集相同或近似的分布。

属性子集选择的基本方法包括以下几种。

1. 逐步向前选择

对于原属性集 S，和 S 的一个初始为空的子集 S'，做下面的循环操作：从 S 中选择最好的属性（最相关的属性）α，加入到 S' 中，直到满足结束条件。结束条件可以有多种，如满足属性个数，满足相关度阈值等。

2. 逐步向后删除

在原属性集 S 上做下面的循环操作：从 S 中选择最坏的属性（最不相关的属性）β，删除这个属性，直到满足结束条件。结束条件与向前选择方法相同。向前选择和向后删除方法的效果应该是相同的。

3. 向前选择和向后删除结合

同时使用向前选择和向后删除方法，每一次选择一个最好的属性，并删除一个最坏的

属性。

4. 判定树(dicision tree)归纳

如图 4-4 所示,判定树像是一棵倒立的树,每棵判定树有一个树根,多个叶子结点和内部结点。

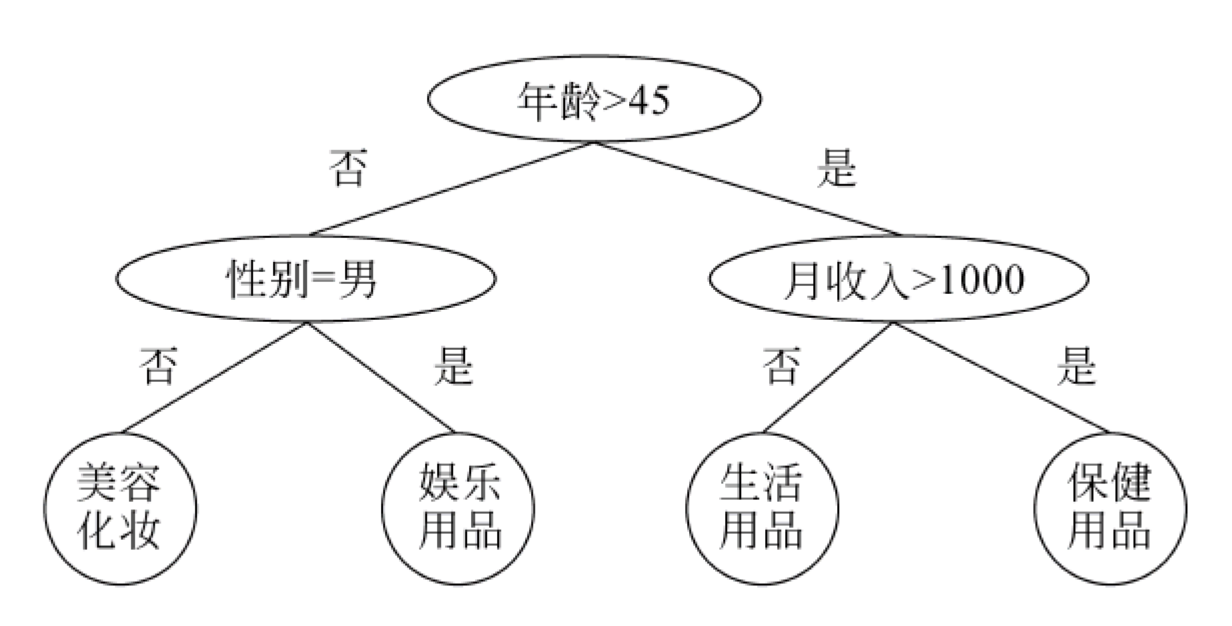


图 4-4　用判定数进行属性归约

每个内部结点表示在一个属性上的测试,每个分支代表一个测试结果输出,从根到每个叶子结点所经过的路径代表该叶子结点满足的测试条件,每个叶子结点代表一个判定类。

判定树算法是一种分类算法,在每一个测试点,算法从属性集中选择相关性最强的属性作为判定条件,根据判定结果把数据划分成两个互斥的类,算法结束时,所有内部结点代表的属性被认为是相关属性而选中,不在树中的属性被认为是不相关的,应该删除。如图 4-4 中,经过判定树归约后的最终属性集为(年龄,性别,月收入)。

5. 基于统计分析的归约

统计分析中的一些算法,如主成分分析(principal component analysis,PCA)、逐步回归分析、公共因素模型分析等可以直接用于维归约。统计分析算法的特点是用少量的特征元组去描述高维的原始知识基。

4.4.4　数据压缩

数据压缩就是用数据编码或者变换,得到原始数据的压缩表示。数据压缩可以减少数据存储而不影响数据挖掘的结果。

数据压缩的方法分为两类:无损压缩和有损压缩。无损压缩如基于熵的编码方法,有损压缩如主成分分析法,是将分散在一组变量上的信息集中到某几个综合指标(主成分)上的探索性统计分析方法。以便利用主成分描述数据集内部结构,实际上也起着数据降维的作用。

主成分分析法具有变差最优性、信息损失最小性、相关最优性和回归最优性,是数据压缩和多元降维的重要工具,其主要思想是要从 n 个属性中找到 k 个最能代表数据特征的 n 维正交向量($k<n$),创建一个由具有“最主要特征”的向量组成的集合来替换原数据,把原数据映射到一个较小的空间,实现数据压缩。

4.4.5　数值归约

数值归约就是通过某种方法,选择较小的数据来替代原数据,减少数据量。常用的方法有直方图、聚类、抽样、回归和对数模型等,其中一些技术在前面介绍过。

1. 直方图(histogram)

直方图技术是一种常用的归约技术,它使用分箱方法对数据进行近似。每个箱代表一个区域范围内的值,箱的宽度代表值域范围,箱的高度代表这个范围内的值的个数,即频率。每个箱可以代表一个属性的值和频率,称为一维直方图,也可以代表两个以上属性的值和频率,称为多维直方图。如果每个箱只表示一个属性值,则称为单桶。

箱的划分已经在第 4.2.2 小节中介绍过,下面是一个用直方图进行归约的例子。

40 组客户交易数据中购买某商品的数量(经过排序)为:1,1,1,1,2,2,2,2,2,4,4,4,5,5,5,5,5,5,5,5,5,8,8,8,10,10,10,10,10,10,12,12,12,12,15,15,15,15,15。

用单桶直方图可以清楚的表示出数据的分布情况,如图 4-5 所示。

根据图 4-5,数据可以被表示成为一些数对。由此可知,单桶直方图具有一定的归约效果,如果希望更进一步的归约数据,可以使每个箱子代表一个值域范围(采用第 4.2.3 小节介绍过的分箱方法分箱),如采用等宽方法分箱,把值域区间划分为 1～5、6～10、11～15 三个范围,得到图 4-6 所示的归约结果。

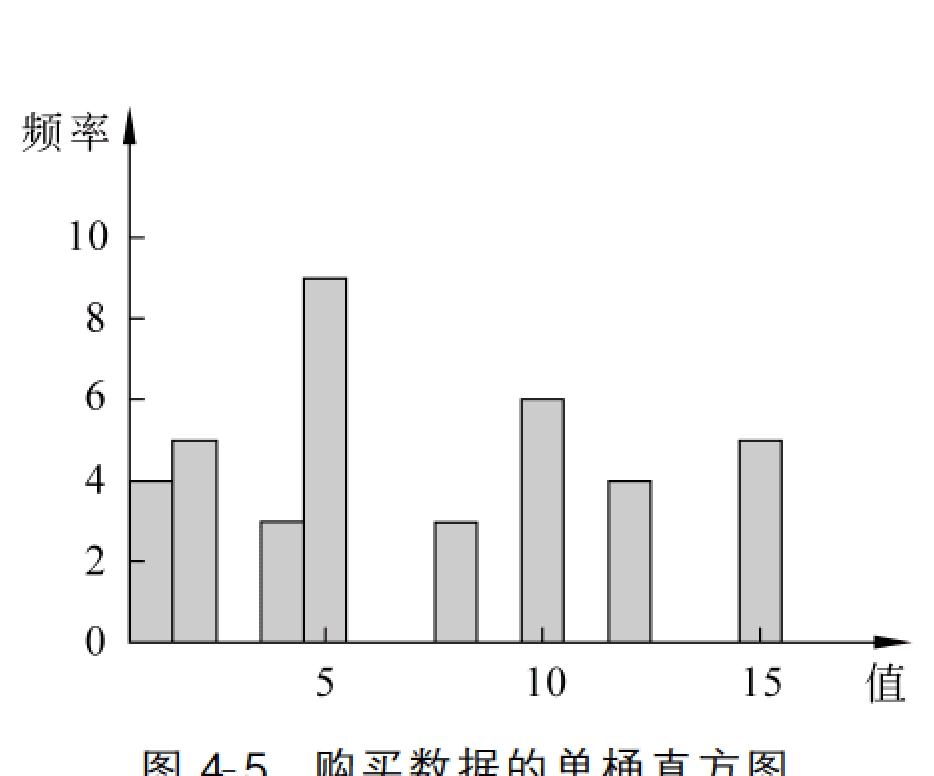


图 4-5 购买数据的单桶直方图

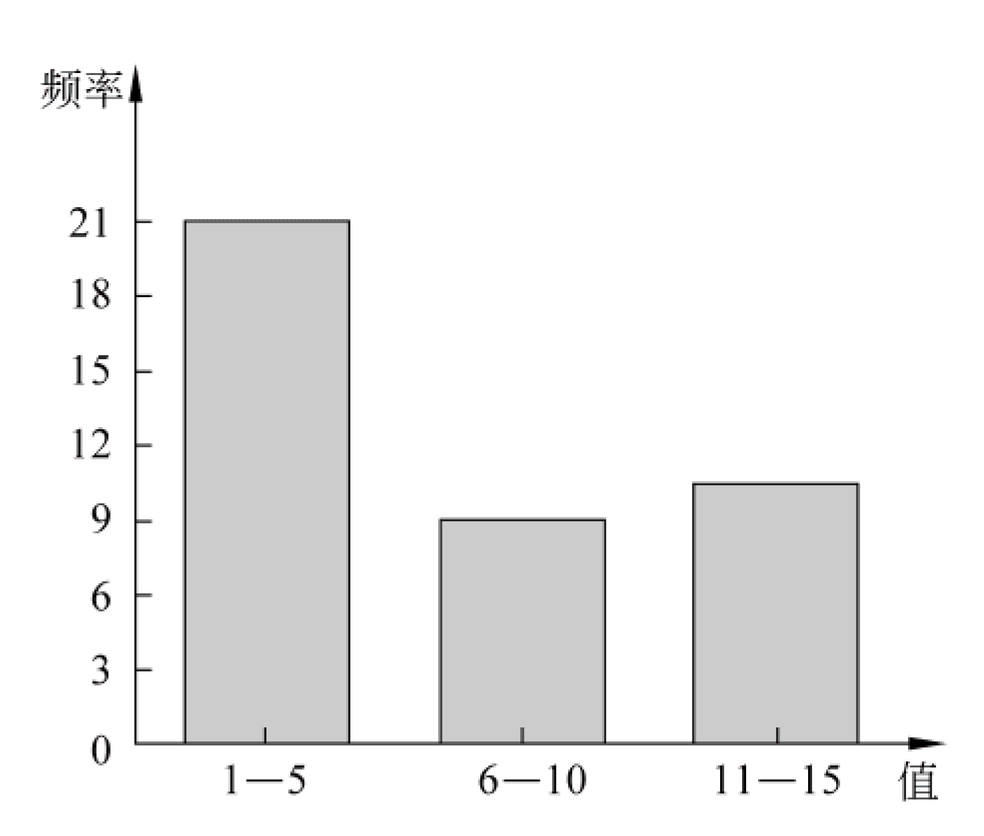


图 4-6 购买数据的等宽直方图(箱宽 5)

2. 聚类

用数据的聚类来代表实际数据。当数据中存在聚类特征时,即数据可以形成有限个聚类时,此方法可以很好的归约数据。聚类方法的基本概念已经在第 4.2.3 小节中介绍过,具体的方法将在后续章节中介绍。

3. 抽样(sampling)

与其他数据归约方法不同,抽样不是对属性进行选择或者删除,它是对记录进行选取,即用较小的数据样本集表示大的数据集。如果一个数据集所包含的记录过多,可以从中抽取一个子集,这个子集就称为样本,样本应该与原数据集具有相同的数据分布。样本的大小一般没有特别的规定,要凭经验和需要确定,但太小的样本因为容易包含太多的偏差而不具有代表性。下面介绍几种抽样的方法。

(1) 不放回简单随机抽样。假定数据集 D 共有 N 条记录,从这 N 条记录中抽取出 K 个样本。不放回抽样即每次抽取一条记录,被抽取的记录不再放回原数据集 D,再做下一次抽样。不放回简单随机抽样对每个记录被抽取的概率都是相等的,为 K/N。

(2) 放回简单随机抽样。从数据集 D 的 N 条记录中做放回抽样,抽取出 K 个样本。放回抽样即每次抽取出一条记录,记录该记录的信息,然后把它放回记录集 D,再做下一次

抽样。在放回抽样中，一条记录可能被抽取 0 次、1 次或多次。

（3）聚类抽样。首先把数据集 D 的数据放入 M 个聚类，然后从每个聚类中抽取样本。

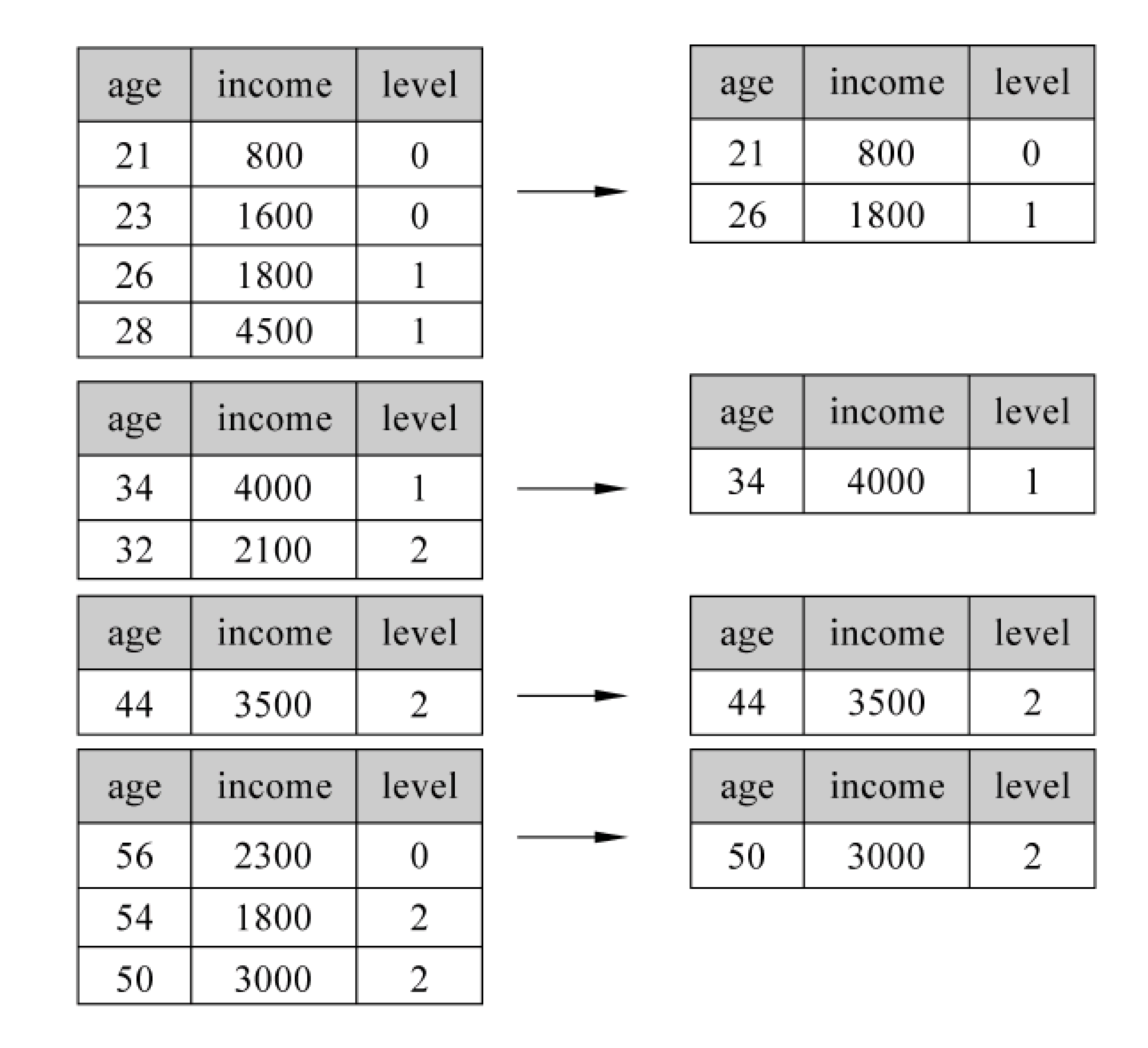
（4）分层抽样。分层即把数据集 D 划分成互不相交的部分，每一部分称为一层。分层抽样就是在数据集 D 的每个层上作简单随机抽样。聚类抽样和分层抽样都可以保持样本的特征不会丢失。

如图 4-7 所示的数据集 D，包含了顾客的年龄、收入和会员等级 3 个属性信息。

使用聚类或者分层方法，可以先根据 age 字段进行聚类或分层，然后再在每个聚类或层中抽取，这样就可以保持 40～49 年龄段之间的信息。如采用按年龄段分层方法，把数据集分为 20～30、30～40、40～50 和 50 岁以上 5 个层，从每个层中随机抽取样本，结果见图 4-8 所示。

age	income	level
21	800	0
56	2300	0
34	4000	1
32	2100	2
54	1800	2
50	3000	2
26	1800	0
23	1600	1
28	4500	1
44	3500	2

图 4-7　示例数据集



age	income	level
21	800	0
23	1600	0
26	1800	1
28	4500	1

age	income	level
21	800	0
26	1800	1

age	income	level
34	4000	1
32	2100	2

age	income	level
34	4000	1

age	income	level
44	3500	2

age	income	level
44	3500	2

age	income	level
56	2300	0
54	1800	2
50	3000	2

age	income	level
50	3000	2

图 4-8　用户数据按年龄分层抽样

在 40～50 年龄段中只有一条记录，如果采用简单随机采样，该记录很有可能不被抽取，而按照年龄分层后采样，则一定可以被抽取，保留了该年龄段的信息。

4. 线性回归

以上介绍的几种抽样方法都是直接从数据集中抽取实际的数据，形成一个样本集，样本集中保存的是样本的实际数据。而线性回归和非线性回归方法用数据模型近似数据，它们并不保存实际数据，而是产生一个数据模型，只保存数据模型的参数，所以也称为参数方法。这类方法只对数值型数据有效。

线性回归的概念已经在第 4.2.3 小节介绍过，根据数据的特征，可以确定一个线性模型来近似数据，并根据实际数据计算出回归系数，形成预测模型。有了这个模型，就可以只保存回归系数和自变量，在需要时根据它们得到因变量的值。

5. 非线性回归

在自变量与因变量之间的关系不是线性关系时，即非线性关系时，要采用非线性回归方法。可以通过一定的函数转换，将非线性关系转换为线性关系，从而采用线性回归分析方法，来解决非线性关系。

一元回归分析可以用来对某些非线性关系进行估计，只要这些非线性关系可以通过取对数变成线性关系。比较常见的非线性关系以及对应的线性模型有以下两种。

(1) $y=e^{a+bx}$，其对数性模型为

$$\ln y = a + bx \tag{4-9}$$

可以用最小二乘法分两步对上述模型进行估计：

先通过运行 $y'=a+bx$ 对 a，b 进行估计，式中 $y'=\ln y$。

然后用式 $y=e^{y'}$ 进行预测。

(2) $y=ab^x$，其对数线性模型为

$$\lg y = \lg a + x\lg b \tag{4-10}$$

$$y' = A + Bx \tag{4-11}$$

其中，$y'=\lg y$，$A=\lg a$，$B=\lg b$，用最小二乘法对上述模型进行估计，计算出参数 A 和 B，y 可以通过式 $y=10^{A+Bx}$ 计算。

4.4.6 离散化与概念分层生成

为了适应算法或者存储的需要，有时需用有限数量的离散数据替代连续数据。通常采用的方法是把数据划分区间，每个区间中的数据用一个值来代替。分箱、直方图、聚类等都是离散化技术，如果在数据集上递归地使用某种离散化技术，就形成了数据集的概念分层。如对数据集 D 递归的使用等宽分箱技术，形成的概念分层如图 4-9 所示。

数据集 D=(0,2,2,5,5,5,10,10,11,14,14,14,18,18,18,26,26,26,26,26,
35,35,35,35,38,38,39,42,42,42,43,43,55,55,55,55,58,60,
66,66,66,69,72,72,73,75,75,75)

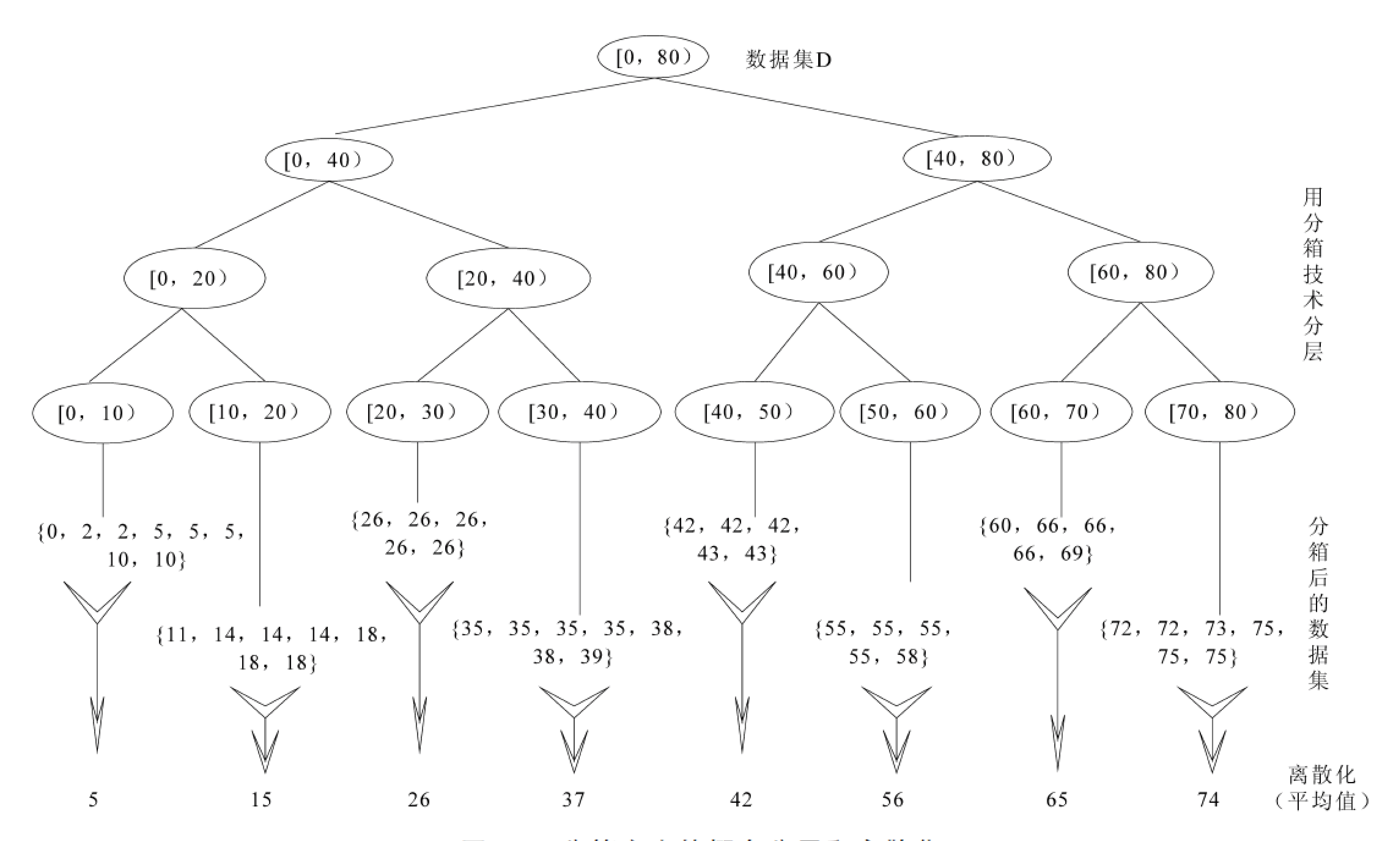


图 4-9 分箱产生的概念分层和离散化

图 4-9 中,树状结构有 4 个层次的结点,根结点表示原始数据集合,其他每一层结点共同表示一个概念分层,它们具有同一个概念级别,而不论处于哪一个概念层中,每个结点都代表一个符合一定条件的数据集合。图中给出了所有叶子结点的数据集合以及对它们离散化后的值,读者可以试着写出其他各层的数据集合以及离散化结果。

在具体应用中,可以根据需要确定分层终止条件,如预先设定层数,或者设定达到的最小区间范围。由这个例子可以看出,概念分层可以用作数据离散化的方法,同时它也可以用作数据归约的方法,下面较详细地介绍离散化与概念分层的方法。

1. 数值数据的离散化与概念分层生成

数值数据的概念分层可以通过数据分析自动产生,这些方法包括前面介绍过的分箱、直方图、聚类,基于熵的离散化(利用熵来递归地划分时间间隔,直到得到的值满足一定精度要求。那么可以用划分间隔和对应的熵值来重新构建原来的数据集)等。它们能够无干预地完成对属性的概念分层,但是这些方法划分出来的层并不考虑边界值是否直观或自然。通常,用户更希望分层具有自然的,易于记忆的、符合人类思维习惯的边界。例如人们希望看到[20,30],而不愿意看到[23.333,36.97]之类的分层。

下面介绍一种通过自然划分分段的方法进行概念分层的过程。该方法应用 3-4-5 规则,递归地将给定数据区域划分为 3、4 或 5 个等宽的区间,3-4-5 规则的具体描述如下。

(1) 如果待划分的区间在最高位上包含 3、6、7 或 9 个不同的值,则将该区间划分成 3 个区间。其中,如果是 3、6 或 9,则划分成等宽的 3 个区间,如果是 7,则按 2-3-2 划分成 3 个区间。

(2) 如果待划分区间最高位上包含 2、4 或 8 个不同的值,则把它划分成 4 个等宽的区间。

(3) 如果待划分区间最高位上包含 1、5 或 10 个不同的值,则把它划分成 5 个等宽的区间。

在每个区间上递归地应用 3-4-5 规则,生成数据的概念分层,直到满足预先设定的终止条件。

图 4-10 表示的是一个用 3-4-5 规则构造概念分层的例子。数据集 D 是某公司每月利润增长数据,数据单位为千元,取值范围在 $-13 \sim 32$ 之间,对最大最小值在 10(千元)上取整,得到一个区间(−20,40),这个区间就是应用 3-4-5 规则的区间。

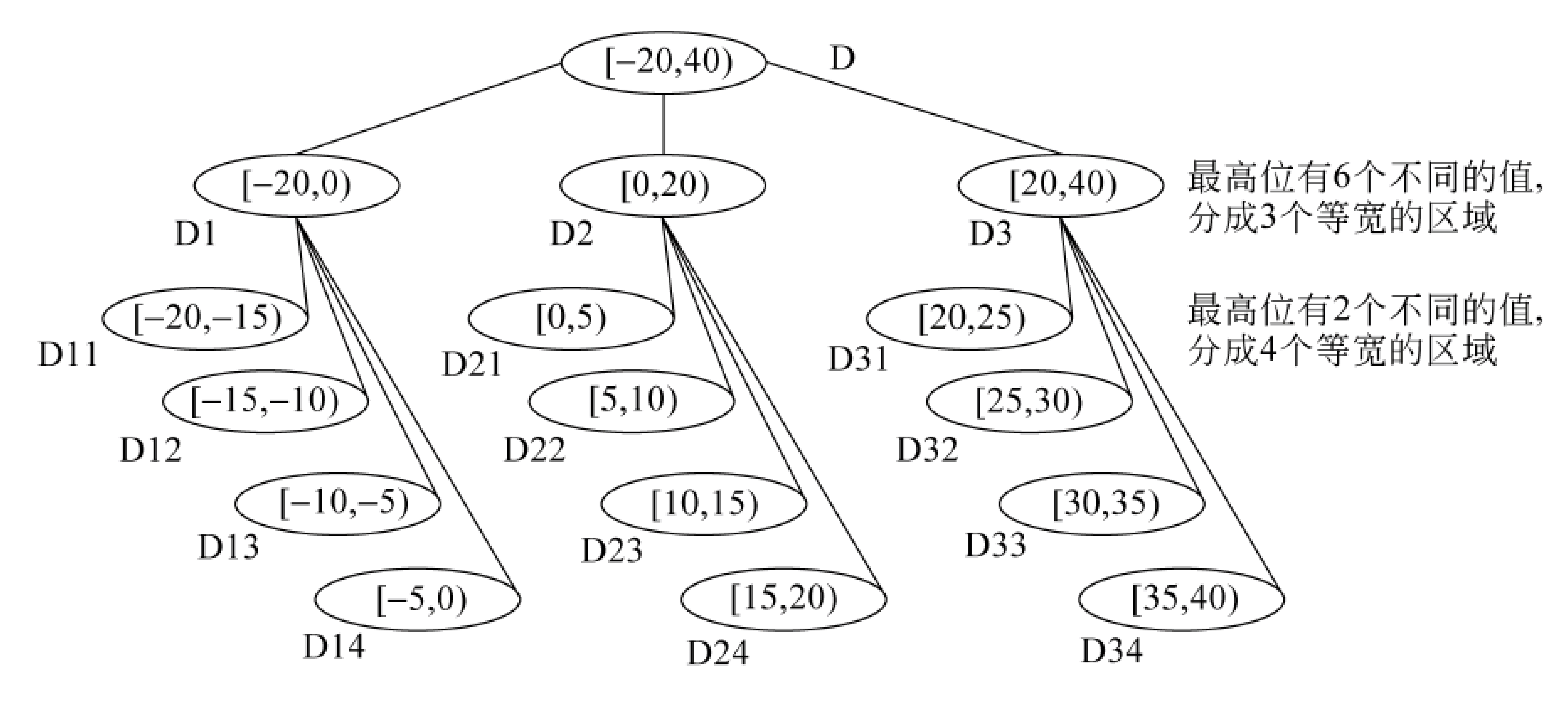


图 4-10 3-4-5 规则产生的概念分层

考察区间[－20,40),最高位有 6 个不同的取值：－2、－1,0,1,2,3,根据 3-4-5 规则,把数据集 *D* 划分为 3 个等宽的区间 D1 、D2 和 D3 ,取值区间分别为[－20,0)、[0,20)和[20,40)。这 3 个等宽的区间最高位分别包含两个不同的取值－2、－1,0、1 和 2、3,所以划分成 4 个等宽的区间,D1 划分为 D11 、D12 、D13 和 D14 ,D2 和 D3 也相同。

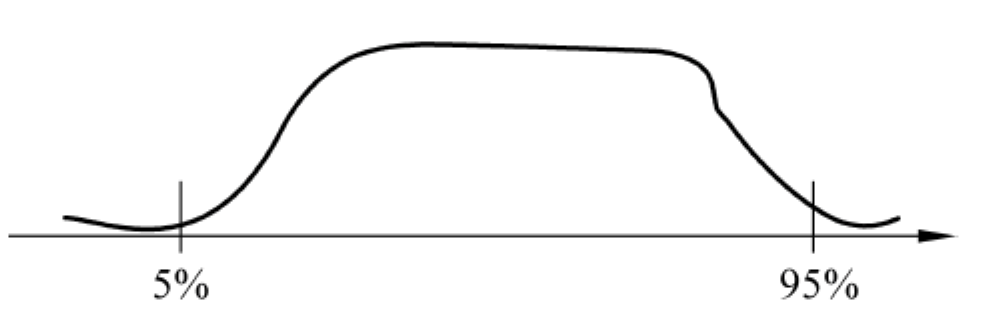


图 4-11　数据集 *D* 的分布曲线

如果数据集 D 的分布曲线呈现图 4-11 所示的情况,区间两端的值所占的比例非常少,可以根据情况设置一个置信区间(如 5%～95%),以这两个点上的值作为初始划分的区间,如[－9,28],同样在 10(千元)上取整,得到区间[－10,30],则第一层划分情况如图 4-12 所示。

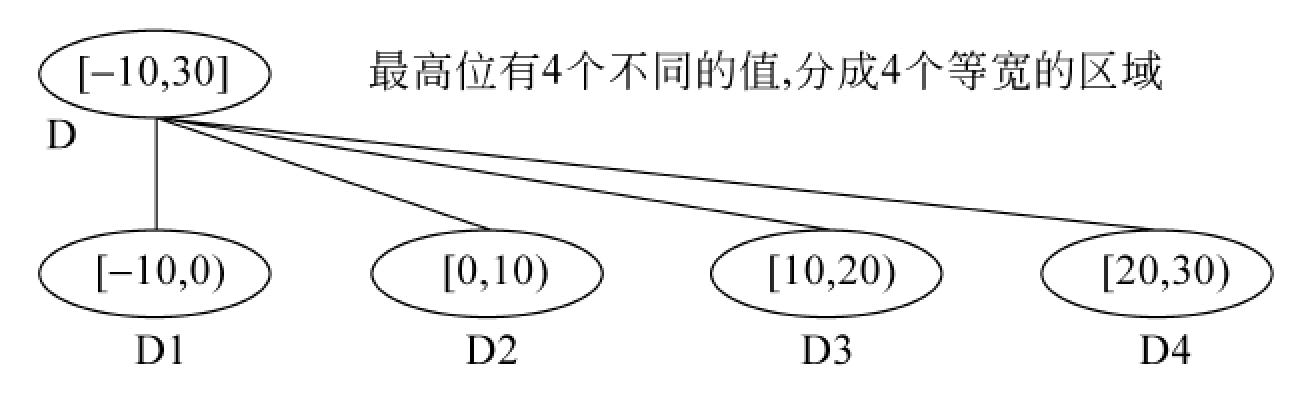


图 4-12　在置信区间 [5%，95%]上的第一层划分

可以看到,由于设置了置信区间[5%,95%],实际上集合 D1 的左边界和 D4 的右边界分别是－10 和 30,不包含集合 D 的实际边界－13 和 32,所以应该在两端补充两个集合表示缺失的数据，如图 4-13 所示。

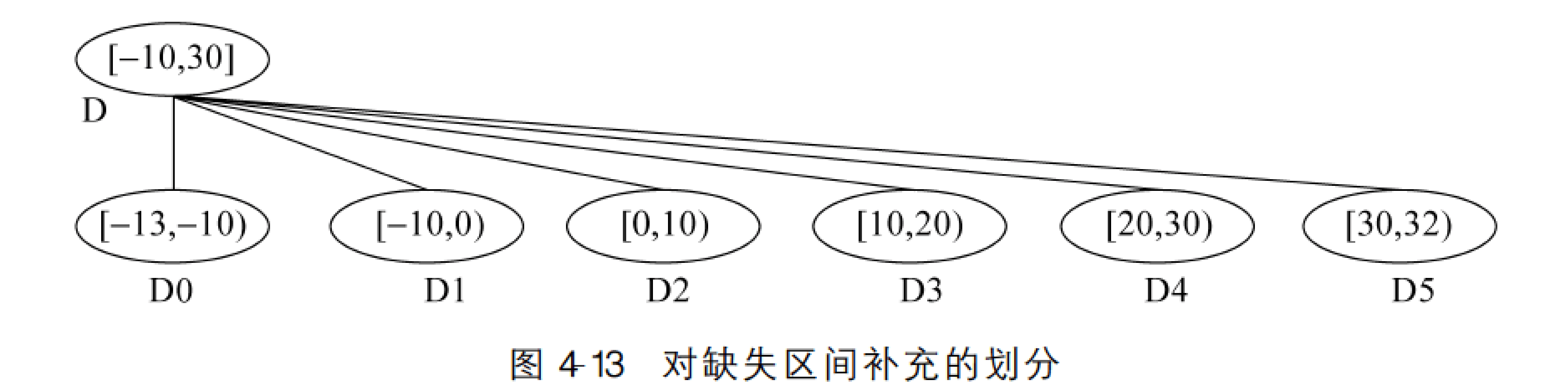


图 4-13　对缺失区间补充的划分

对区间 D0～D5 应用 3-4-5 规则,得到的分层如图 4-14 所示。

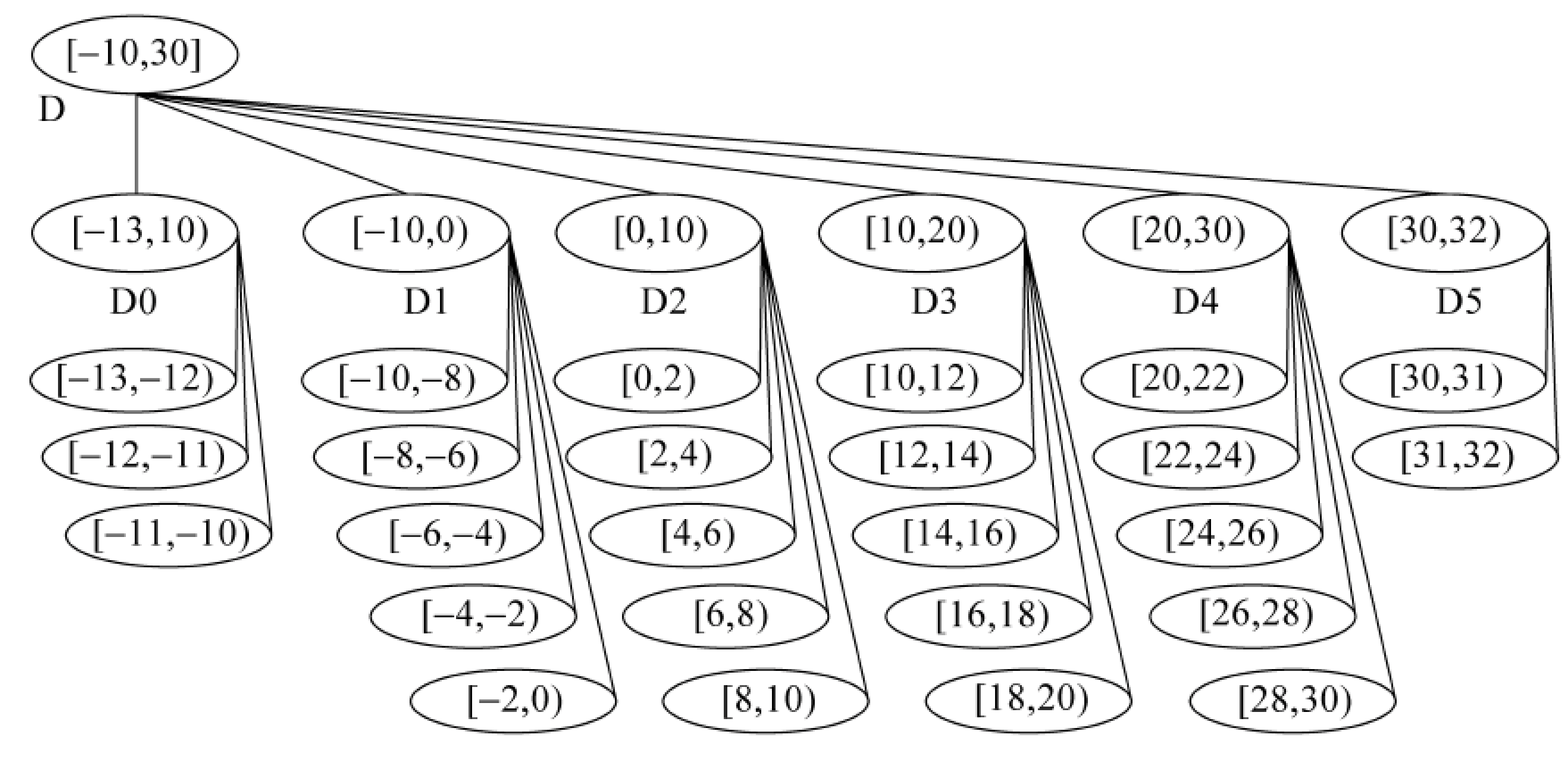


图 4-14　对图 4-13 进一步分层

可以递归地划分下去，直到满足一定的要求，如区间大小达到预定的阈值。

2. 分类数据的概念分层生成

分类数据是指分类属性值所包含的数据（可以是数值型、字符型或字符串等），所谓分类属性，就是指那些具有有限个取值的属性，如商品类型、店铺的位置和客户类型等，这些数据之间没有大小关系，所以不能采用数值数据的分层方法。下面介绍几种典型的用于分类数据概念分层的方法。

（1）由用户或专家在模式级显式地说明数据的包含关系。如果分类属性之间存在部分包含或者完全包含的关系，可以由用户或者领域专家说明属性之间的包含关系，根据这个包含关系形成概念分层。如，在属性组：year，month，day 之间就存在完全包含关系：day⊂month⊂year，可以用这个关系定义它们的概念分层，如图 4-15（a）所示。

year
month
day
(a)

year
month
Ten days
day
(b)

图 4-15　对属性组：year，month，day 的概念分层

（2）通过显式数据分组说明分层结构的一部分。在产生包含关系的属性之间，有的属性取值数目较少，而有一些属性包含大量不同的离散值，对取值过多的属性，通过枚举值逐个定义概念分层是不可能的，而对于值较少的属性，可以进行手工的分组。如把日期值{1，2，3，4，5，6，7，8，9，10}定义为“上旬”，{11，12，13，14，15，16，17，18，19，20}定义为“中旬”，{21，22，23，24，25，26，27，28，29，30}定义为“下旬”，就形成图 4-15(b)所示的概念分层。

（3）根据属性值的个数自动产生分层。只给出属性组，不定义属性的包含关系，根据属性值的个数自动产生分层。此方法的根据是：与定义在较低概念层的属性相比，定义在较高概念层的属性通常具有较少数量的不同的值，把具有最少不同值的属性放在最高层，属性的不同值数目越多，所处的概念层越低。并不是所有的属性之间的关系都可以这样确定，也有特殊的情况，如上面的例子中，如果属性 year 的不同取值个数超过 12，则会产生类似：day⊂year⊂month 的概念分层，所以有时候需要对自动产生的分层进行手工调整。

（4）根据数据语义产生分层。有时候，由于用户对数据结构认识的误差，或者操作上的失误，只提供了相关属性组的部分属性，不能形成一个完整的分层结构，此时就要借助于数据语义，即在数据模式中加入属性的说明，这些说明把属性组联系在一起。当一个属性被增加进属性组时，依靠数据语义可以把所有相关的属性增加进来。

小结

数据预处理是数据挖掘中非常重要的一个环节，挖掘使用的数据来源于实际操作数据源，这些数据源可能是多个数据库，其结构和规则可能是不同的，这将导致原始数据非常的杂乱、不可用。即使在同一个数据库中，也可能存在重复、不完整、大量的空缺值和不一致现象。预处理的目的就是为数据挖掘提供干净一致的数据，本节从数据清洗、数据集成、数据变换和数据归约几个方面介绍了预处理的方法。数据清洗涉及属性选择与处理、空缺值处理、噪声数据处理和不平衡数据处理；数据集成涉及在集成过程中的模式匹配问题、数据冗余问题和数值冲突等问题的解决方法；数据变换讨论了数据的平滑、聚集、概化和规范化方

法；数据规约主要介绍了数据立方体聚集、维归约、数据压缩、数值压缩、离散化和概念分层生成及其具体的方法。通过学习本章内容，可以了解数据预处理的重要性，了解原始数据中存在的问题，掌握各种预处理方法。

习题 4

1. 列举实际业务操作数据中存在的问题以及这些问题产生的原因。

2. 数据预处理涉及哪些方法，这些方法分别用于解决数据中的哪方面的问题？

3. 说明属性选取的原则。

4. 说明填补空缺值的方法和这些方法的优缺点。

5. 下面是一个超市某种商品连续 24 个月的销售数据(百元)：

21,16,19,24,27,23,22,21,20,17,16,20,23,22,18,24,26,25,20,26,23,21,15,17

使用统一权重、统一区间、和自定义区间方法对数据分箱，做出各种分箱方法得到的直方图。

6. 对上题中分箱后的数据采用平均值、边界值或中值等方法进行平滑。

7. 如果挖掘算法需要把第 5 题中的商品销售数据规范化到区间[0,1]上，采用最小一最大规范化方法，请写出规范化后的结果。

8. 试采用一种分箱方法，对以下某种商品连续 30 周的销售利润数据进行归约(千元)：

3,2,5,7,4,2,5,6,8,8,4,5,4,6,2,3,7,5,5,4,6,3,4,7,8,3,6,4,2,3

9. 解释本章中提到的几种数据抽样方法。

10. 用等宽分箱技术对排序后的数据集 D=(0,0,2,2,2,4,8,8,8,12,12,12,12,15,15,16,16,16,16,21,21,21,25,25,25,25,25,28,28,29,34,34,34,34,37,37,44,44,44,58,58,58,58,58,63,63,66,66,66,69,74,74,74,78,78)进行离散化，使得每箱宽度不大于 5，形成概念分层。

11. 对连续数值型数据集 D，取值范围为 0～70，试用 3-4-5 规则对其进行离散化。

第 5 章　关联规则方法

数据挖掘中许多常用的传统模式发现技术，如决策树、分类规则和聚类技术都属于机器学习领域的研究成果。而关联规则挖掘不同，该技术的出现极大扩展了数据挖掘的研究，有着巨大的影响，成为数据挖掘研究的一个重要分支。由于关联规则可以有效地发现数据之间的重要关联关系，并且规则的表达形式简洁，易于解释和理解，从大型数据库中挖掘关联规则的问题已经成为近年来数据挖掘研究领域中的一个热点。

采用关联规则挖掘类似于“90％的顾客在买面包和黄油的同时也会买牛奶”这样的知识，可以帮助决策者确定市场经营策略。关联规则在很多其他场合也有成功的应用。例如，在商业销售行为中，关联规则可用于确定交叉销售策略，以得到更高的收入；在保险业务方面，如果出现了不常见的索赔要求组合，则可能为欺诈，需要作进一步的调查；在医疗方面，可以发现某一类患者的共同特征，或者可找出可能的治疗组合及观察各种组合的治疗效果；在银行方面，对顾客进行分析，可以推荐感兴趣的服务，等等。由于这些实际应用目标的差异，在关联规则大的理论框架下有许多面向实际应用目标的理论和方法等待探索和创新。

5.1　关联规则的概念和分类

关联规则（association rules）概念产生于 1993 年，其最初的目的是为了寻找大量商务数据库中项集之间的有趣联系，由 Agrawal、Imielinski 和 Swami 提出。

5.1.1　关联规则的概念

关联规则用来发现在同一事件中出现的不同项的相关性，即找出事务中频繁发生的项或属性的所有子集，以及项目之间的相互关联性。为了方便描述，首先说明以下符号。

D：事务数据库（database）；

I：项目（item）集合，$I=\{i_1,i_2,\cdots,i_m\}$，其中，$i_1,i_2,\cdots,i_m$ 为数据库中的项目；

T：数据库中的事务（transaction）；

X：项集（itemset），即项目的集合；

k 项集：包含 k 个项目的集合；

支持度 $s(X)$：项集 X 的支持度，表示数据库中包含项集 X 的交易数据的条数；

频繁项集：也称为频繁模式，指支持度大于用户指定的最小支持度的项集；

频繁 k-项集（k frequent itemset）：长度为 k 的频繁项集。

在以上符号基础上，对关联规则的形式化描述如下：

设项目集合 $I=\{i_1,i_2,\cdots,i_m\}$由 m 个不同的项目组成，D 为事务数据库，D 中的每一个事务 T 是 I 的一个子集，即 $T\subset I$。一个项目的集合称为项集，包含 k 个项目的集合称为 k 项集，项集 X 的支持度，记为 $s(X)$，表示包含该项集的交易数据的条数，如果一个项集的支持度大于用户指定的最小支持度（min_sup），则称它是频繁的，长度为 k 的频繁项集称为频繁 k-项集，一

个频繁项集也称为频繁模式。关联规则是形如 $A \Rightarrow B$ 的蕴涵式，其中 $A \subset I, B \subset I$，并且 $A \cap B = \varnothing$。规则 $A \Rightarrow B$ 的支持度 $s(A \Rightarrow B)$ 定义为 D 中包含 $A \cup B$ 的事务所占的百分比，表示项集 $A \cup B$ 在 D 中出现的概率，$s(A \Rightarrow B) = |\{T: A \cup B \subseteq T\}| / |D|$。

规则 $A \Rightarrow B$ 的置信度 c(min_con)定义为 D 中包含项集 $A \cup B$ 的事务数和包含项集 A 的事务数的比值，表示当项集 A 出现时，项集 B 出现的概率，$c(A \Rightarrow B) = s(A \cup B) / s(A)$，置信度大于用户指定的最小置信度值的规则是可信的。

关联规则挖掘的任务是找到事务数据库 D 中支持度和置信度分别满足用户指定的最小支持度 min_sup 和最小置信度 min_con 的规则 $A \Rightarrow B$。

关联规则挖掘问题分为两个子问题(或者说是两个步骤)：

(1) 找出 D 中所有的频繁项集；

(2) 从频繁项集中产生关联规则。

其中，第一个子问题所需要的计算量和磁盘 I/O 量都较大，几乎所有的关联规则挖掘算法都是针对第一个子问题提出的。

5.1.2 关联规则的分类

可以从不同角度对关联规则进行分类，下面介绍几种最常见的分类方法。

1. 基于规则中处理的变量类别分类

基于关联规则中处理的变量类别，可以分为布尔型和数值型两种。

布尔型关联规则处理的值都是离散的、种类化的，关联规则显示这些变量之间的关系；数值型关联规则是对数值型字段进行处理，将其进行动态的分割，或者直接对原始的数据进行处理，当然数值型关联规则中也可以包含种类变量。

例如：性别="男"=>职业="网络工程师"，是布尔型关联规则；性别="男"=>avg(收入)=3500，其中的收入项是数值类型，所以是一个数值型关联规则。

2. 基于规则中数据的抽象层次分类

基于规则中数据的抽象层次分类，可以分为单层关联规则和多层关联规则。

在单层的关联规则中，所有的变量都不考虑现实数据具有多个不同层次的特点；在多层的关联规则中，考虑数据的多层性。

例如：Sony 数码照相机=>Sony 彩色喷墨打印机，是一个细节数据上的单层关联规则；数码照相机=>Sony 彩色喷墨打印机，是一个较高层次和细节层次之间的多层关联规则。

3. 基于规则中涉及的数据维数分类

基于规则中涉及的数据维数分类，关联规则可以分为单维和多维。

在单维关联规则中，只涉及数据的一个维，如用户购买的商品；多维的关联规则中，要处理的数据将会涉及多个维。或者说，单维关联规则是处理单个属性中的某些关系；多维关联规则是处理多个属性之间的某些关系。

例如：啤酒=>尿布，这条规则只涉及用户购买的商品；性别="男"=>职业="网络工程师"，这条规则就涉及两个字段的信息，是一条两维关联规则。

4. 基于模式与规则之间的相互关系分类

基于模式与规则之间的相互关系分类，可以分为完全频繁模式挖掘、最大频繁模式挖掘

和闭合频繁模式挖掘。

由于应用环境和目的不同，在以上多种关联规则挖掘方法中，一维单层布尔型关联规则挖掘方法是其他方法的基础。

5.2 Apriori 算法

已有的一维单层布尔型关联规则频繁项集发现方法可以分为产生候选项集和不产生候选项集两类。产生候选项集的方法最初由 Agrawal 等人于 1994 年提出，这就是著名的 Apriori 算法，这个算法成为后来绝大多数关联规则挖掘算法的基础。而不产生频繁项集的挖掘方法最典型的是 FP-Grwoth 方法。

Apriori 命名的由来是因为算法使用了频繁项集性质的先验知识，即 Apriori 性质。Apriori 性质的内容是：频繁项集的所有非空子集也都必须是频繁的。这个性质被用于减少候选频繁项集的数量。Apriori 算法将发现关联规则的过程分为两步：第 1 步是通过迭代，检索出源数据中的所有频繁项集，即支持度不低于用户设定阈值的项集；第 2 步是利用第 1 步中检索出的频繁项集构造出满足用户最小置信度的规则。

5.2.1 产生频繁项集

Apriori 算法产生频繁项集采用迭代方法实现。每一次迭代包括两个步骤：产生候选项集；由候选项集中产生频繁项集。称第 k 次迭代产生的候选项集为候选 k 项集，记为 C_k，第 k 次迭代产生的频繁项集为频繁 k 项集，记为 L_k。第 k 次迭代产生的项集长度为 k。

产生频繁项集的过程可以用下面的算法表示：

```
L1 = {frequent items};
for (k=1;Lk !=∅; k++) do
begin
    Ck+1 = apriori-gen(Lk);
    for each transaction t∈ T do
        begin
            Ct = subset(Ck+1, t);
            for each candidate c∈ Ct do
                c.support++;
            end
        Lk+1= {c∈ Ct+1 | c.support> =min_sup}
end
return ∪k Lk;
```

Apriori 算法的计算过程如下：

1. 求频繁 1 项集 L_1

以项目集合 I 作为候选 1 项集 C_1，扫描数据库一次，统计各个项目的出现次数，根据设定的最小支持度得出频繁 1 项集 L_1；

2. 求频繁 $k+1$ 项集 L_{k+1}（即执行 apriori-gen(L_k)）

第 1 步，对前 $k-1$ 个项目相同的每两个 k 频繁模式执行 join 操作，得到候选 $k+1$ 项集 C_{k+1}；

```
insert into Ck+1
select p.item1,p.item2,…,p.itemk,q.itemk from Lk.p,Lk.q
where p.item1=q.item1 and ... and p.itemk-1=q.itemk-1 and p.itemk<q.itemk
```

第 2 步，根据 Apriori 性质，对 C_{k+1} 进行剪枝。

```
for each itemset c∈ Ck+1 do
for each k-subsets 属于 c   do
    if(s 不包含于 Lk) then delete c from Ck+1
```

扫描数据库一遍，确定每个 $c \in C_{k+1}$ 的支持计数，据此得出频繁 k 项集 L_{k+1}。

在第 1 次迭代的第 1 步中，产生包含所有 1-项集的候选集，即源数据中所有的项。通过对事务的搜索，计算出支持度。然后，选择支持度大于所需阈值的 1-项集为频繁项集。这样，通过第 1 次迭代，得到所有频繁 1-项集。

在第 $k(k>1)$ 次迭代中，对 L_{k-1} 中的项集两两进行连接操作，然后对得到的项集，根据 Apriori 性质判断每个项集是否是可能的频繁项集，得到 C_k，再根据 C_k 的支持度确定出 L_k。直到不能够再从 C_k 产生 L_{k+1}，频繁项集的计算过程结束。通过 k 次迭代，就可以产生长度从 1 到 k 的所有频繁项集。

5.2.2 产生频繁项集的实例

假定某数据库 D 中包含有项目{I1}、{I2}、{I3}、{I4}和{I5}，用户要求的最小支持度阈值 $s=20\%$。数据库 D 如图 5-1 所示。

事务数据库

TID	项目列表
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I3,I4
T6	I1,I3
T7	I1,I2,I3,I5
T8	I2,I3,I4
T9	I2,I3,I5
T10	I3,I5

图 5-1 示例数据库

1. 第 1 次迭代，产生频繁 1-项集

按照上述产生频繁项集的过程，在进行第 1 次迭代时，首先产生候选 1-项集 C_1，如表 5-1(a)。然后，算法计算每一个候选项集的出现次数，计算支持度，如表 5-1(b)。最后，选择支持度 $s \geq 20\%$ 的项目，生成频繁 1-项集 L_1，如表 5-1(c)。

表 5-1 Apriori 算法的第 1 次迭代

候选 1-项集 C_1
{I1}
{I2}
{I3}
{I4}
{I5}

(a)

候选 1-项集	计数	s[%]
{I1}	4	40
{I2}	7	70
{I3}	7	70
{I4}	4	40
{I5}	4	40

(b)

频繁 1-项集 L_1	计数	s[%]
{I1}	4	40
{I2}	7	70
{I3}	7	70
{I4}	4	40
{I5}	4	40

(c)

2. 第 2 次迭代，产生频繁 2-项集

在 Apriori 算法中，使用 $L_1 * L_1$ 产生候选项集。“*”运算定义为：

$$L_k * L_k = \{X \cup Y, X, Y \in L_k, |X \cap Y| = k-1\}$$

当 $k=1$ 时，该运算为单连接。设 C_2 为包含在第 2 次迭代中产生的 2-项集。按照上述

公式，产生的 2-项集数量应该为$|L_1| * (|L_1|-1)/2$。在此例中为 $5*4/2=10$。因此，产生 10 项候选 2-项集 C_2，如表 5-2(a)。然后，算法统计每一个候选集的出现次数并计算支持度，如表 5-2(b)。最后，选择支持度 $s \geqslant 20\%$的项集形成频繁 2-项集 L_2，如表 5-2(c)。

表 5-2　Apriori 算法的第 2 次迭代

候选 2-项集 C_2
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I4}
{I3,I5}
{I4,I5}

(a)

候选 2-项集	计数	s[%]
{I1,I2}	3	30
{I1,I3}	2	20
{I1,I4}	1	10
{I1,I5}	2	20
{I2,I3}	4	40
{I2,I4}	3	30
{I2,I5}	3	30
{I3,I4}	2	20
{I3,I5}	3	30
{I4,I5}	0	0

(b)

频繁 2-项集 L_2	计数	s[%]
{I1,I2}	3	30
{I1,I3}	2	20
{I1,I5}	2	20
{I2,I3}	4	40
{I2,I4}	3	30
{I2,I5}	3	30
{I3,I4}	2	20
{I3,I5}	3	30

(c)

3. 第 3 次迭代，产生频繁 3-项集

候选项集 C_3 可以使用运算 $L_2 * L_2$ 产生。连接运算可产生项集{I1,I2,I3}，{I1,I2,I5}，{I1,I3,I5}，{I2,I3,I4}，{I2,I3,I5}，{I2,I4,I5}，{I3,I4,I5}，而其中项集{I2,I4,I5}和{I3,I4,I5}的子集{I4,I5}不包含在频繁 2-项集中，根据 Apriori 性质，这两个项集不能够成为候选 3-项集，其他 5 个项集的所有子集都是频繁的，所以可以成为候选 3-项集。由此得到的候选 3-项集 C_3 如表 5-3(a)所示。对 C_3 的计数结果在表 5-3(b)中，最终得到如表 5-3(c)所示的频繁 3-项集 L_3。

表 5-3　Apriori 算法的第 3 次迭代

候选 3-项集 C_3
{I1,I2,I3}
{I1,I2,I5}
{I1,I3,I5}
{I2,I3,I4}
{I2,I3,I5}

(a)

候选 3-项集	计数	s[%]
{I1,I2,I3}	1	10
{I1,I2,I5}	2	20
{I1,I3,I5}	1	10
{I2,I3,I4}	1	10
{I2,I3,I5}	2	20

(b)

频繁 3-项集 L_3	计数	s[%]
{I1,I2,I5}	2	20
{I2,I3,I5}	2	20

(c)

根据 Apriori 性质，如果要求一个频繁项集的所有子集都是频繁的，则 L_3 中至少要有 3 个项集，才可能产生一个频繁 4-项集，因为一个频繁 4-项集至少有 3 个长度为 3 的子集。在此例中 L_3 无法产生候选 4-项集，所以算法到此停止迭代。

Apriori 算法不仅计算所有频繁项集的支持度，也计算那些在由 Apriori 性质不能排除的非频繁候选项集的支持度。所有这些非频繁，但符合 Apriori 性质的候选项集的集合被称为负边界。如果项集是非频繁的，但它的所有子集都是频繁的，那么它就在负边界中。

在上述例子中，从表 5-1 和表 5-2 可以看出，负边界由项集{I1，I4}，{I4，I5}，{I1，I2，I3}，{I1，I3，I5}和{I2，I3，I4}组成。

负边界在一些 Apriori 的改进算法中更为重要，例如生成频繁项集或者导出负关联规则时，提高其有效性。

5.2.3　从频繁项集产生关联规则

在使用 Apriori 算法或其他类似的算法所发现的所有频繁项集基础上，挖掘关联规则。对任意一个频繁项集，首先计算它的子集，如对频繁 3-项集{I1，I2，I5}，可以计算得到它的子集{I1，I2}和{I5}，{I1，I5}和{I2}，{I2，I5}和{I1}，由频繁项集产生的规则可知，这些子集都是频繁的。可以得到规则{I1，I2}→I5，{I1，I5}→I2，{I2，I5}→I1。然后，计算规则的置信度，例如：$c(\{I1,I2\}\rightarrow I5)=s(I1,I2,I5)/s(I1,I2)=2/3$，置信度 c 大于给定的阈值的规则就是强关联规则。

在上述例子中，如果设定规则的置信度为 60%，得到的部分强关联规则如表 5-4 所示。

表 5-4　Apriori 算法得到的强关联规则

频繁项集	产生的规则	置信度	强关联规则	置信度
{I1，I2}	I1→I2	3/4	I1→I2	3/4
	I2→I1	3/7		
{I1，I3}	I1→I3	2/4		
	I3→I1	2/7		
{I1，I5}	I1→I5	2/4		
	I5→I1	2/4		
{I2，I3}	I2→I3	4/7		
	I3→I2	4/7		
{I2，I4}	I2→I4	3/7		
	I4→I2	3/4	I4→I2	3/4
{I2，I5}	I2→I5	3/7		
	I5→I2	3/4	I5→I2	3/4
{I3，I4}	I3→I4	2/7		
	I4→I3	2/4		
{I3，I5}	I3→I5	3/7		
	I5→I3	3/4	I5→I3	3/4
{I1，I2，I5}	I1，I2→I5	2/3	I1，I2→I5	2/3
	I1，I5→I2	2/2	I1，I5→I2	2/2
	I2，I5→I1	2/3	I2，I5→I1	2/3
{I2，I3，I5}	I2，I3→I5	2/4		
	I2，I5→I3	2/3	I2，I5→I3	2/3
	I3，I5→I2	2/3	I3，I5→I2	2/3
(a)	(b)		(c)	

值得注意的是，并不是所有被挖掘出来的强关联规则都有意义或者都有用，需要注意它们是否有负关联的情况。这个问题，在此不做详细分析。

5.3 FP-Growth 算法

不产生候选项集的方法中，最有代表性的是 FP-Growth 算法。FP-Growth 算法采用一种称为频繁模式树(FP-tree)的结构，FP-tree 是为了存储与频繁模式相关的关键信息而设计的一棵压缩的、扩展前缀树结构。FP-growth 算法包括构成 FP-tree 和从 FP-tree 得到频繁模式两个阶段。为了构成 FP-tree 需要扫描数据库两次，第一次统计事务中所有项的出现次数，并按照出现次数的大小排序，形成一个列表(假设所有事务中包含的项都以这个顺序排列)。第二次扫描数据库构建 FP-tree。树的根结点是一个空结点，不代表任何信息，根结点之外的所有结点都代表一个项目，用数对(项目名：支持数)表示。为了方便对树的遍历，还采用了一个称为项目头的表格结构，表中按出现频率递减的顺序存放了所有项目，并且保存了项目在树中出现位置的指针，同一项目在树中的多次出现形成一个结点链。

由 FP-tree 得到频繁模式的过程是从频度最小的频繁项开始，采用一种称为 FP-Growth 的方法自底向上地在条件模式库上进行挖掘。

由于不产生大量的候选集，FP-Growth 算法的计算时间远远小于 Apriori(大约一个数量级)，但是在处理长模式时，构造 FP-tree 的代价较高。到目前为止，FP-tree 是最杰出的关联规则挖掘算法。

5.3.1 FP-Growth 算法计算过程

FP-tree 的构建算法描述如下：

(1) 扫描事务数据库一次，得到频繁项的集合 F 及其支持度。对 F 按支持度降序排列，生成频繁项列表 L_1。

(2) 创建 FP-tree 的根结点 T，以“null”标记。对于数据库中的每条事务，执行操作 3～5。

(3) 将事务中的频繁项目按 L_1 中的次序排列。排序后的频繁项表表示为 $[p|P]$，其中 p 是第一个频繁项，P 是剩余项目列表。

(4) 调用 insert-tree($[p|P]$，T)，即由根结点 T 开始，如果 T 有子结点 N 满足 N.item-name$=p$.item-name，则结点 N 的计数增 1；否则创建一个新结点 N，将其计数置为 1，连接到其父结点 T，并且通过结点链结构将其连接到具有相同 item-name 的结点。

(5) 如果频繁项表 P 非空，递归地调用 insert-tree(P，N)。

图 5-2 是一个示例数据库，图 5-3 表示了从示例数据库创建的 FP-tree。

TID	项目列表
T1	I1,I2,I5
T2	I2,I3
T3	I2,I4
T4	I1,I2,I4
T5	I1,I4,I5
T6	I2,I3
T7	I3,I4
T8	I1,I2,I3,I5
T9	I1,I2,I3

图 5-2 示例数据库

从 FP-tree 挖掘频繁模式的方法称为 FP-Growth，描述如下：

```
Procedure FP-growth(Tree,σ)
if Tree 含单个路径 p then
    for 路径 p中结点的每个组合 (记作 β)
    产生模式 β∪σ,其支持度为 β 中结点的最小支持度；
```

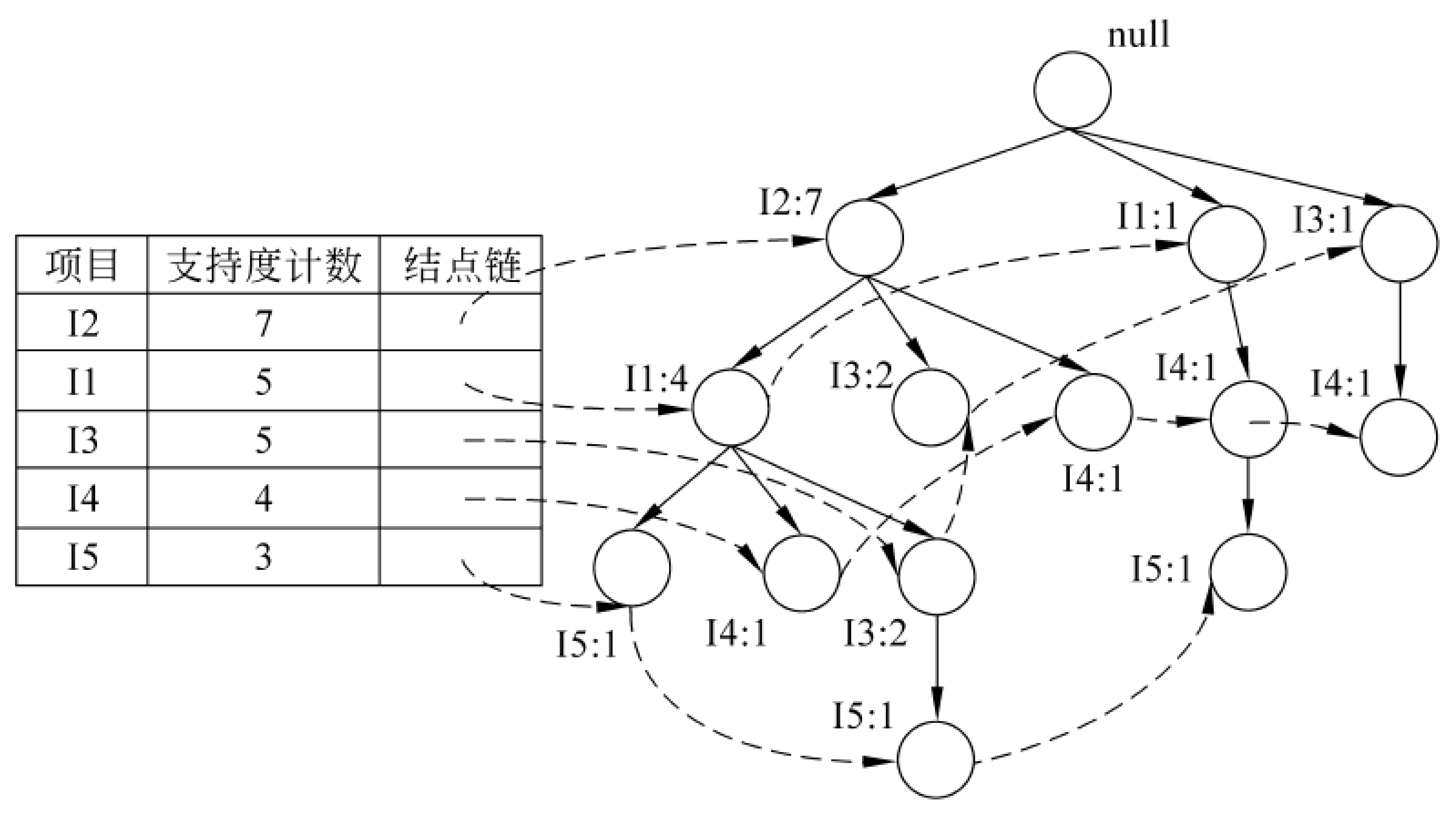


图 5-3 FP-tree

else for each σ_i 在 Tree 的头部

{

产生一个模式 $\beta=\sigma_i \cup \sigma$,其支持度为 σ_i 的支持度;

构造 β 的条件模式基,然后构造 β 的条件 FP-Tree;

if Tree$\neq\varnothing$ then

调用 Fp-growth(Tree,β)

}

算法将发现长模式的问题转换成递归地发现一些短模式,然后连接后缀。使用最不频繁的项作为后缀,降低了搜索的开销。

5.3.2 FP-Growth 算法示例

第 1 步,扫描图 5-2 所示的示例数据库一次,统计各个项目的支持度计数,将各个频繁项目按支持度递减排序,形成频繁项列表,即头表。

第 2 步,再次扫描数据库,按照第 5.3.1 小节中 FP-tree 创建算法将数据库中的事务信息压缩到 FP-tree 上,结果如图 5-3 所示。

第 3 步,由支持度最小的项目 I5 开始,按照 I5 的同名结点链找到树上所有名为 I5 的结点,计算每个 I5 到根的路径,得到项目 I5 的条件模式基:<I2,I1:1>,<I2,I1,I3:1>,<I1,I4:1>,合并这三个条件模式基得到 I5 的条件模式树。合并的结果得到两个分支:<I2:2,I1:2,I3:1>和<I1:1,I4:1>,假设设定的最小支持度为 2,则这两个分支可以构成条件模式树<I2:2,I1:2>,因为在分支<I2:2,I1:2,I3:1>中,I3 的支持度只有 1,不能成为条件模式树上的结点。而分支<I1:1,I4:1>的支持度也为 1,同样不能够作为条件模式树上的分支。最终构成 I5 的条件模式树只有 1 个分支,如图 5-4 所示。

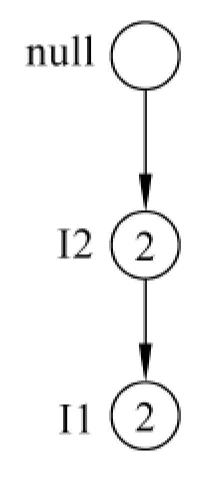


图 5-4 I5 的条件模式树

由条件模式树,可以得出的频繁模式为<I2,I5:2>,< I1,I5:2>,< I2,I1,I5:2>,生成了所有以 I5 为后缀的频繁模式。

按照步骤 3 的过程可以依次计算得出以 I4,I3,I1,I2 为后缀的频繁模式。

FP-Growth 类算法的缺点是构建 FP-tree 的时间和空间代价较高，特别是在挖掘数据集为稀疏数据的时候，效率甚至低于 Apriori 算法；另外，基于 FP-tree 结构的算法在挖掘过程中都只记录频繁项信息，同 Apriori 算法及其改进算法相比，在应用领域上有所限制。

5.4 利用 SQL Server 2005 进行关联规则挖掘

SQL Server 2005 提供了比 SQL Server 2000 更多的数据挖掘模型，包括关联规则、聚类分析、决策树、逻辑回归、神经网络、时序、顺序分析和聚类分析、线性回归等。这些功能由 Analysis Services 服务器提供，集成在 SQL Server Business Intellegence Development Studio 中。

下例为某银行的客户贷款数据库，其中记录了客户的背景数据以及贷款情况，银行需要从这些数据中发现客户背景与不良贷款之间的关系，也就是要发现具有哪些背景的用户更容易产生不良贷款。解决这一问题，可以采用关联规则挖掘方法，从现有用户数据中发现形如“背景特征＝＞贷款情况”的规则，并将规则应用于以后的贷款业务中。本节应用 SQL Server 2005 的数据挖掘工具实现这一目的。

1. 数据准备

图 5-5 和图 5-6 是两张数据表，分别保存客户基本情况和贷款余额情况。

表 - dbo.客户基本情况表 | 表 - dbo.贷款余额表 | 摘要

客户代码	客户名称	客户类型	经济性质	隶属关系	法人资格	客户状态	重点标志
77040007000045	K045单位	物资	国有	区县属	法人	正常	非重点
77020130000017	K017单位	其他	其他	地州市属	法人	正常	非重点
77160001000082	K082单位	商业	集体	其他	法人	正常	非重点
77020108000050	K050单位	商业	国有	中央	法人	正常	一级重点
77120002000057	K057单位	其他	其他	地州市属	法人	正常	非重点
77020110000088	K088单位	工业	国有控股	省属	法人	正常	国家重点
77020110000179	K179单位	工业	国有	省属	法人	正常	一级重点
77020110000185	K185单位	工业	国有	省属	法人	正常	一级重点
77020240000019	K019单位	物资	其他	省属	授权法人	正常	一级重点
77020110000080	K080单位	其他	部队	军办	法人	正常	非重点
77070004000025	K025单位	工业	国有	地州市属	授权法人	半停产	一级重点
77170001000134	K134单位	其他	国有	地州市属	法人	正常	二级重点
77070008000009	K009单位	工业	股份合作	区县属	法人	正常	非重点
77040004000002	K002单位	工业	国有控股	省属	法人	正常	一级重点
77040020000002	K002单位	供销	国有	省属	授权法人	正常	一级重点
77050004000023	K023单位	工业	国有控股	省属	法人	正常	一级重点
77170009000012	K012单位	工业	国有	区县属	法人	正常	非重点
77070004000027	K027单位	工业	私营	地州市属	法人	正常	非重点
77170005000012	K012单位	工业	国有	区县属	法人	正常	非重点
77180006000007	K007单位	工业	国有	区县属	法人	正常	非重点

图 5-5　客户基本情况表

首先，为了区别每一笔业务，在“贷款余额表”中添加一个主键列，命名为“业务号”，使用“smallint”数据类型并设置为标志字段。

其次，为了简化挖掘过程，将这两张表的信息合并到同一张表中。合并方法采用图 5-7 所示的 SQL 语句实现。

数据合并之后产生的新表命名为 t_dm，其中包含了客户基本信息表的所有信息和贷款余额表中的部分信息。如图 5-8 所示。

2. 实现挖掘任务

使用 SQL Server 2005 挖掘工具提供的关联规则挖掘模型对表 t_dm 中的数据进行挖

表 - dbo.客户基本情况表 | 表 - dbo.贷款余额表 | 摘要

分行代码	客户代码	业务发生日	余额	正常	关注	次级	可疑	损失
7702	77020101000060	2002-3-20 0:00:00	44000000.00	44000000.00	0.00	0.00	0.00	0.00
7702	77020105000009	2002-8-30 0:00:00	2700000.00	0.00	0.00	0.00	2700000.00	0.00
7702	77020105000009	2002-8-30 0:00:00	1200000.00	0.00	0.00	0.00	1200000.00	0.00
7702	77020105000009	2002-7-30 0:00:00	3000000.00	0.00	0.00	0.00	3000000.00	0.00
7702	77020105000009	2002-9-25 0:00:00	500000.00	0.00	0.00	0.00	500000.00	0.00
7702	77020105000009	2002-9-25 0:00:00	1410000.00	0.00	0.00	0.00	1410000.00	0.00
7702	77020105000009	2002-8-30 0:00:00	1400000.00	0.00	0.00	0.00	1400000.00	0.00
7702	77020105000010	2002-8-31 0:00:00	526457.00	526457.69	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	10474.00	10474.58	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	70623.00	70623.69	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	20393.00	20393.01	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	468.00	468.83	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	20882.00	20882.30	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	3519.00	3519.27	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	298505.00	298505.28	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	54395.00	54395.64	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	83952.00	83952.70	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	6266.00	6266.48	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	313747.00	313747.75	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	1271511.00	1271511.10	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	450770.00	450770.20	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	560833.00	560833.51	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	21508.00	21508.80	0.00	0.00	0.00	0.00
7702	77020105000010	2002-8-31 0:00:00	444310.00	444310.50	0.00	0.00	0.00	0.00

图 5-6　贷款余额表

GHOST-49F0F...LQuery1.sql* | 表 - dbo.贷款余额表* | 表 - dbo.客户基本情况表

```
select a.*,b.正常,b.关注,b.次级,b.可疑,b.损失,b.业务号
into t_dm
from 客户基本情况表 as a
JOIN 贷款余额表 as b
ON a.客户代码=b.客户代码
```

图 5-7　合并表数据

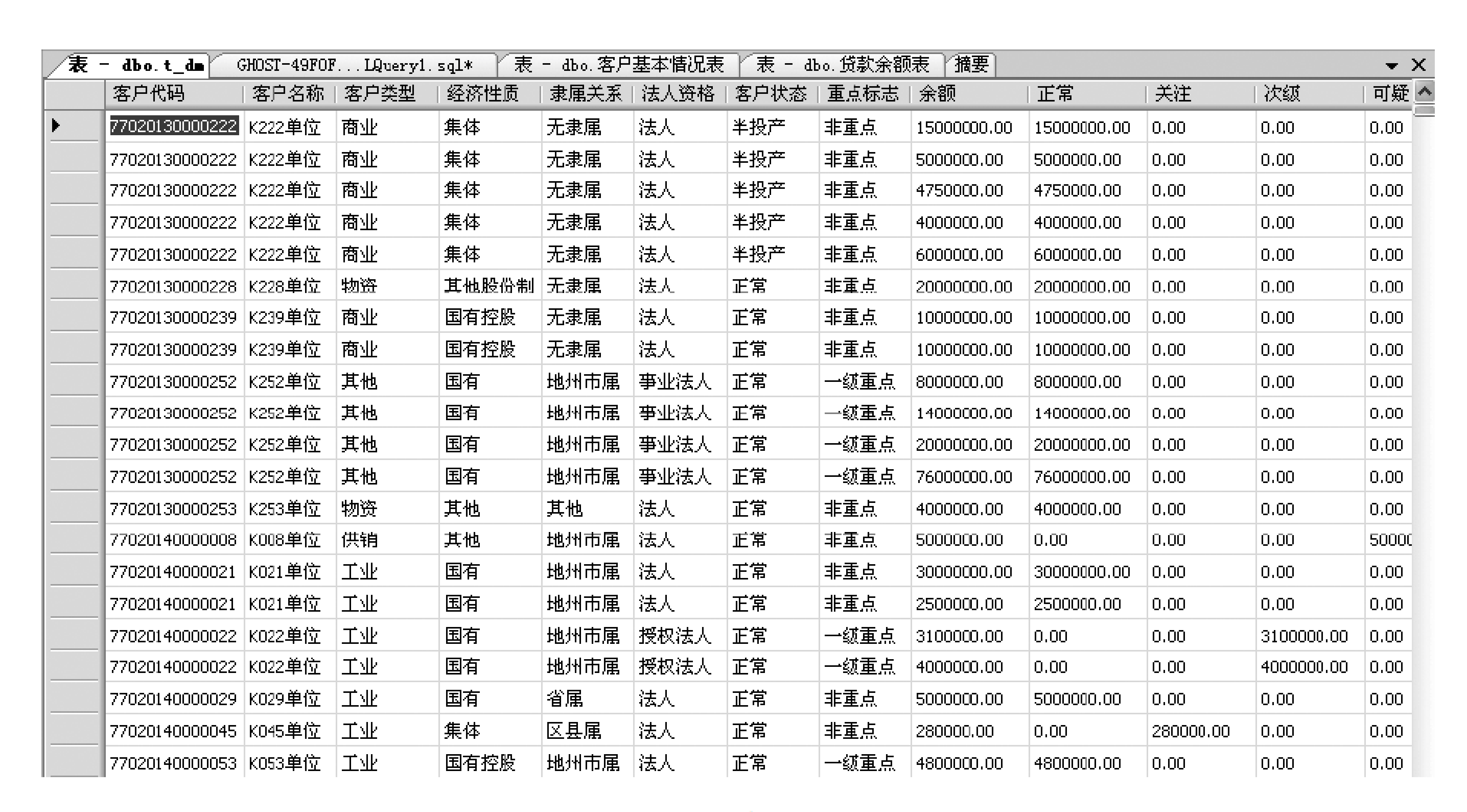


表 - dbo.t_dm | GHOST-49F0F...LQuery1.sql* | 表 - dbo.客户基本情况表 | 表 - dbo.贷款余额表 | 摘要

客户代码	客户名称	客户类型	经济性质	隶属关系	法人资格	客户状态	重点标志	余额	正常	关注	次级	可疑
77020130000222	K222单位	商业	集体	无隶属	法人	半投产	非重点	15000000.00	15000000.00	0.00	0.00	0.00
77020130000222	K222单位	商业	集体	无隶属	法人	半投产	非重点	5000000.00	5000000.00	0.00	0.00	0.00
77020130000222	K222单位	商业	集体	无隶属	法人	半投产	非重点	4750000.00	4750000.00	0.00	0.00	0.00
77020130000222	K222单位	商业	集体	无隶属	法人	半投产	非重点	4000000.00	4000000.00	0.00	0.00	0.00
77020130000222	K222单位	商业	集体	无隶属	法人	半投产	非重点	6000000.00	6000000.00	0.00	0.00	0.00
77020130000228	K228单位	物资	其他股份制	无隶属	法人	正常	非重点	20000000.00	20000000.00	0.00	0.00	0.00
77020130000239	K239单位	商业	国有控股	无隶属	法人	正常	非重点	10000000.00	10000000.00	0.00	0.00	0.00
77020130000239	K239单位	商业	国有控股	无隶属	法人	正常	非重点	10000000.00	10000000.00	0.00	0.00	0.00
77020130000252	K252单位	其他	国有	地州市属	事业法人	正常	一级重点	8000000.00	8000000.00	0.00	0.00	0.00
77020130000252	K252单位	其他	国有	地州市属	事业法人	正常	一级重点	14000000.00	14000000.00	0.00	0.00	0.00
77020130000252	K252单位	其他	国有	地州市属	事业法人	正常	一级重点	20000000.00	20000000.00	0.00	0.00	0.00
77020130000252	K252单位	其他	国有	地州市属	事业法人	正常	一级重点	76000000.00	76000000.00	0.00	0.00	0.00
77020130000253	K253单位	物资	其他	其他	法人	正常	非重点	4000000.00	4000000.00	0.00	0.00	0.00
77020140000008	K008单位	供销	其他	地州市属	法人	正常	非重点	5000000.00	0.00	0.00	0.00	50000
77020140000021	K021单位	工业	国有	地州市属	法人	正常	非重点	30000000.00	30000000.00	0.00	0.00	0.00
77020140000021	K021单位	工业	国有	地州市属	法人	正常	非重点	2500000.00	2500000.00	0.00	0.00	0.00
77020140000022	K022单位	工业	国有	地州市属	授权法人	正常	一级重点	3100000.00	0.00	0.00	3100000.00	0.00
77020140000022	K022单位	工业	国有	地州市属	授权法人	正常	一级重点	4000000.00	0.00	0.00	4000000.00	0.00
77020140000029	K029单位	工业	国有	省属	法人	正常	非重点	5000000.00	5000000.00	0.00	0.00	0.00
77020140000045	K045单位	工业	集体	区县属	法人	正常	非重点	280000.00	0.00	280000.00	0.00	0.00
77020140000053	K053单位	工业	国有控股	地州市属	法人	正常	一级重点	4800000.00	4800000.00	0.00	0.00	0.00

图 5-8　合并产生的表

掘。观察客户的背景特征与贷款情况的关系。

从“开始”菜单启动 Microsoft Visual Studio，如图 5-9 所示。

启动 Microsoft Visual Studio 之后，选择“文件”|“新建”|“项目”菜单，打开新建项目对话框。新建一个 Analysis Services 项目。并且在对话框中指定项目名称和存放位置。见图 5-10。

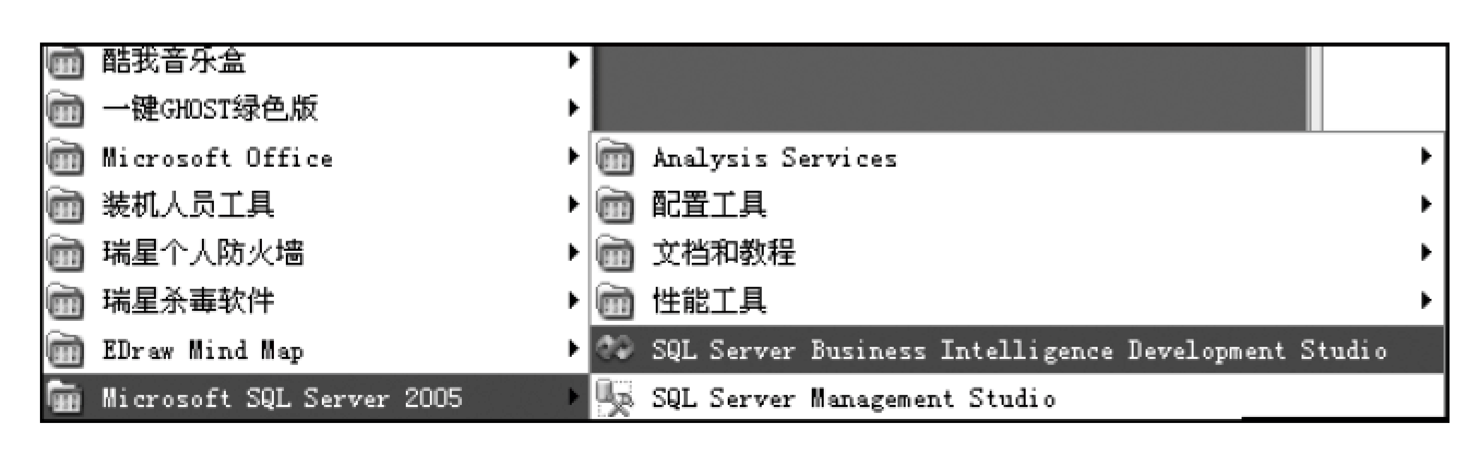


图 5-9　启动 Microsoft Visual Studio



图 5-10　新建 Analysis Services 项目

创建成功之后,将在解决方案资源管理器中显示新建项目,如图 5-11 所示。

使用 SQL Server 的挖掘模型,首先要为项目创建数据源。在 Microsoft SQL Server 2005 Analysis Services(SSAS)中,数据源实际上是一个连接字符串,表示到数据源的连接,用来指明 Analysis Services 如何使用托管 Microsoft.NET Framework 或本机 OLE DB 访问接口连接到物理数据存储区,该连接字符串包含服务器名称、数据库、安全性、超时值以及其他与连接相关的信息。Analysis Services 直接支持多种数据源,包括 Microsoft SQL Server 数据库以及通过其他产品创建的数据库。



图 5-11　Analysis Services 项目信息

创建数据源的过程参见第 2.5 节。这里,创建一个名为 dm 的新数据源,该数据源将连接数据表 t_dm 所在的数据库。

接下来创建新的数据源视图。数据源视图提供一组已经存在的、可浏览的、持久化数据库对象(例如表、视图和关系)。Analysis Services 中的联机分析处理(OLAP)和数据挖掘对象可以引用这些数据库对象。对这些对象进行组织和配置,以便为数据源提供完整的架构表示形式。在 Analysis Services 项目或部署数据库中生成数据源视图后,该数据源视图就可供 Analysis Services 中的任何 OLAP 或数据挖掘对象使用。创建数据源视图的方法同创建数据源相同,可以使用资源管理器中的右键菜单,如图 5-12 所示。

创建数据源视图向导见图 5-13。



图 5-12　新建数据源视图



图 5-13　数据源视图向导

首先为数据源视图选择一个数据源。可以从选择列表中选择一个已有的数据源，也可以使用这个页面中的“新建数据源”按钮新建一个数据源，如图 5-14 所示。



图 5-14　选择数据源

接下来会显示所选数据源所连接的数据库中的表或者视图对象，如图 5-15 所示。

在可用对象中单击分析所需要的表，然后用[>]按钮将选中的表移动到包含对象中，可以选中多个表。如果要删除一个已经选中的表，可以使用[<]按钮将其从“包含对象”中移除。如果要选中或者取消所有表，可以使用[>>]或[<<]按钮。本例中，已经将训练模型的数据放进一个表 t_dm 中，所以将该表加入到“包含对象”中。

筛选器的功能是筛选“可用对象”下列出的对象，类似于“查找”功能。键入目标对象名中包含的字符串，再单击“筛选器”即可列出包含指定字符串的对象名称。筛选器不区分大



图 5-15　选择表和视图

小写。筛选器字符串中的任意位置都可以使用使用通配符：* 和%可以代表任意字符串；?代表一个单个的字符。

单击“显示系统对象”按钮，可以在“可用对象”显示框中显示该数据源中的系统对象。当在“包含对象”中选中一个对象，并单击“添加相关表”按钮时，将会把与该对象相关的对象移动进来，此选项不能添加视图。

选择好数据表之后，单击“下一步”按钮，进入图 5-16 所示的界面。为新建的数据源视图取一个名字 v_dm，就完成了数据源视图的创建。

数据源视图向导
完成向导
请提供一个名称，然后单击“完成”以创建新数据源视图。
名称(A):
v_dm
预览(P):
v_dm
dbo.t_dm
< 上一步(B)
下一步(N) >
完成(F)
取消

图 5-16　命名数据源视图

回到 Microsoft Visual Studio 主界面，就可以看到上面选中的表对象的信息，见图 5-17。如果要创建和使用多维数据集，则参考第 2.5 节创建数据仓库的过程。

下一步创建挖掘结构。挖掘结构和挖掘模型是 SSAS 数据挖掘中使用的两个主要对象。挖掘结构是一种数据结构，它定义生成挖掘模型的数据域。一个挖掘结构可包含多个

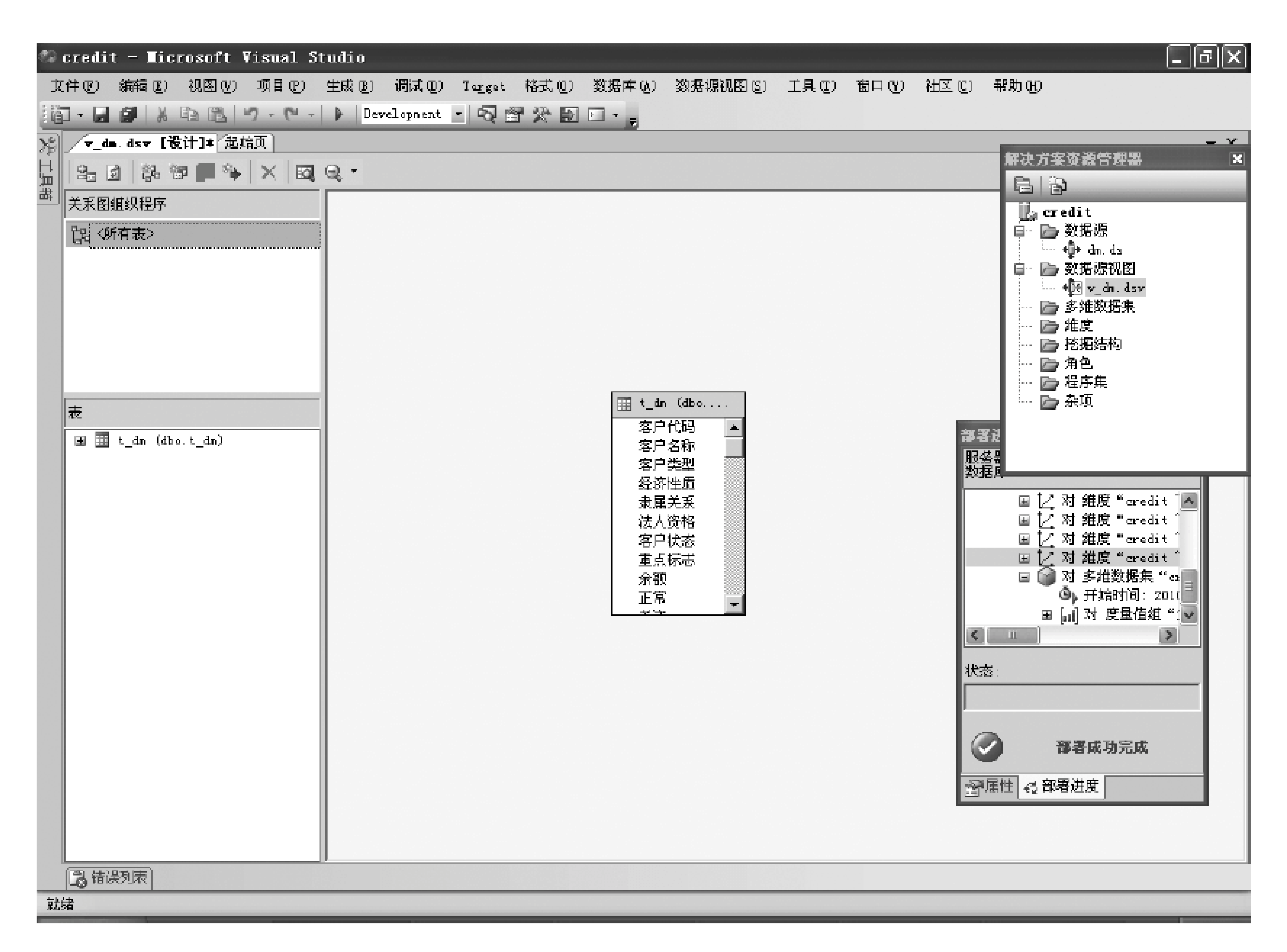


图 5-17　数据源视图具体信息

共享相同域的挖掘模型。挖掘结构对数据源包含的数据进行说明，由挖掘结构列构成，这些列包含列数据类型、列内容类型等信息。挖掘结构也可包含嵌套表，嵌套表表示事例实体与其相关属性之间的一对多关系。例如，如果客户说明信息位于一个表中，而客户贷款信息位于另一个表中，则可使用嵌套表将这些信息组合到一个事例中。客户标识符是实体，采购信息是相关属性。

同挖掘结构一样，挖掘模型也包含列。挖掘模型包含在挖掘结构之内，继承由挖掘结构定义的所有属性值。该模型可以使用挖掘结构包含的所有列，或使用其中一部分列。另外，挖掘模型中还包含使用列的方法和模型所用的算法。

创建挖掘结构也使用右键菜单启动创建向导，如图 5-18 和图 5-19 所示。



图 5-18　新建挖掘结构

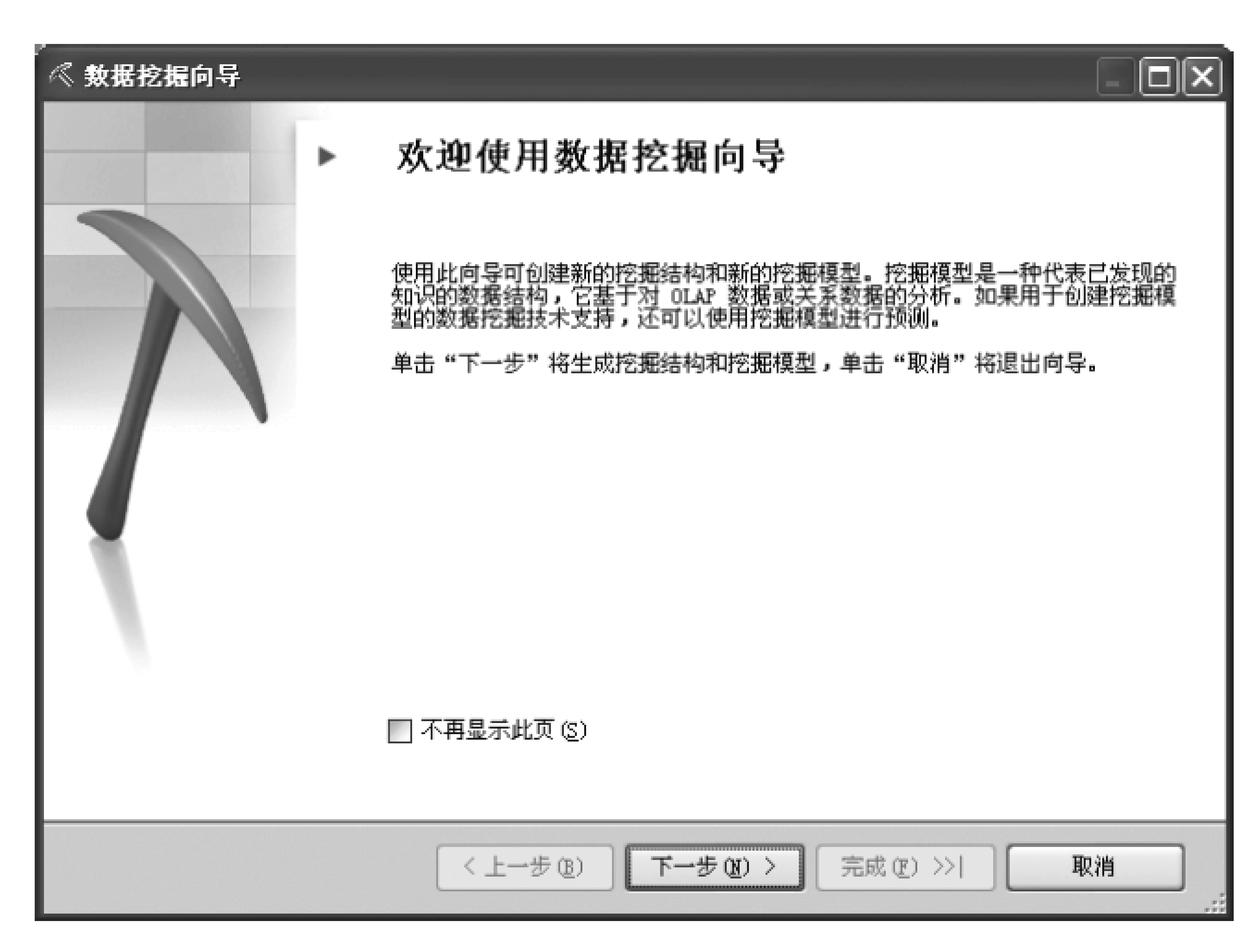


图 5-19　数据挖掘向导欢迎界面

首先，选择创建挖掘结构所用的方法和使用的数据挖掘技术，如图 5-20 和图 5-21 所示。



图 5-20　选择创建挖掘结构所用的方法



图 5-21　选择挖掘结构所用的数据挖掘技术

选择为数据挖掘结构提供数据的数据源视图，如图 5-22，选择上一步创建的数据源视图 v_dm。单击“浏览”按钮可以显示所选数据源视图中包含的数据表信息。

图 5-23 设定所用数据表的类型。选择用于定义挖掘结构的表，并且将其设置为“事例”表或者“嵌套”表。未被选择的表不能用来定义挖掘结构。

本例中，挖掘数据是由客户背景和贷款数据构成的，每条事务包含客户代码(主键)、背景信息和贷款情况，将该表选择为事实表。

选择表类型之后，需要进一步从数据表中挑选数据列，并将数据列指定为“键列”、“输入列”或者“可预测列”，如图 5-24 所示。



图 5-22　选择数据源视图



图 5-23　指定表类型



图 5-24　指定列

可以同时指定一个列为可预测列和输入列。如果将列选为“键”,则表示将该列用作数据的唯一标志。“输入列”指用来预测“可预测列”的列,将来出现在规则的左侧。“可预测列”指被预测的列,将来出现在规则的右侧。本例中,使用“业务号”作为键列;客户基本信息应作为输入列,“客户代码”、“客户名称”一般只用来区分客户,并不具有特殊含义,所以不在模型中选用该列;客户贷款的情况信息作为可预测列。

建立挖掘结构还需要设定数据列的数值形式和数据类型,这通过图 5-25 的界面来设定。创建向导可以根据数据源的特征自动检测到以上内容,也可以手动的设置在下拉列表中选择。可选择的数据值形式包括以下几种。



图 5-25　指定列内容和数据类型

(1) DISCRETE:离散值。离散属性列中的值即使是数值类型,也不意味着是有序数据;这些值之间是明确独立的,且不可能为小数值,如电话区号。

(2) DISCRETIZED:从连续列派生的值,如果列值是连续的,通过分组或者存储桶离散为几个值段。有关数据离散化的信息,可以参见第 4.4 节。

(3) ORDERED:列包含定义有序集的值。所谓有序集,并不表示在该集的值之间存在任何差或量级关系。有序属性列就内容类型而言是离散的。

(4) CYCLICAL:该列包含表示循环有序集的值。例如,一周内顺序编号的 7 天便是循环有序集,因为第 1 天紧跟第 7 天。循环列就内容类型而言既有序又离散。

一般情况下,单击右下方的“检测”按钮,自动检测各个列的取值情况和数据类型。

最后,为挖掘结构指定一个名称,就完成了挖掘结构的创建,如图 5-26 所示。

单击“完成”按钮,回到主界面,此时出现了.dsv 选项卡,如图 5-27 所示。其中包含了挖掘结构、挖掘模型、挖掘模型查看器、挖掘准确性图表和挖掘模型预测。

挖掘结构确定一旦确定,就可以从挖掘结构建立挖掘模型,并对模型进行计算。默认情况下,模型计算根据默认的参数进行,如最小支持度、最小置信度、最大最小项集尺寸等。也



图 5-26　为挖掘结构命名

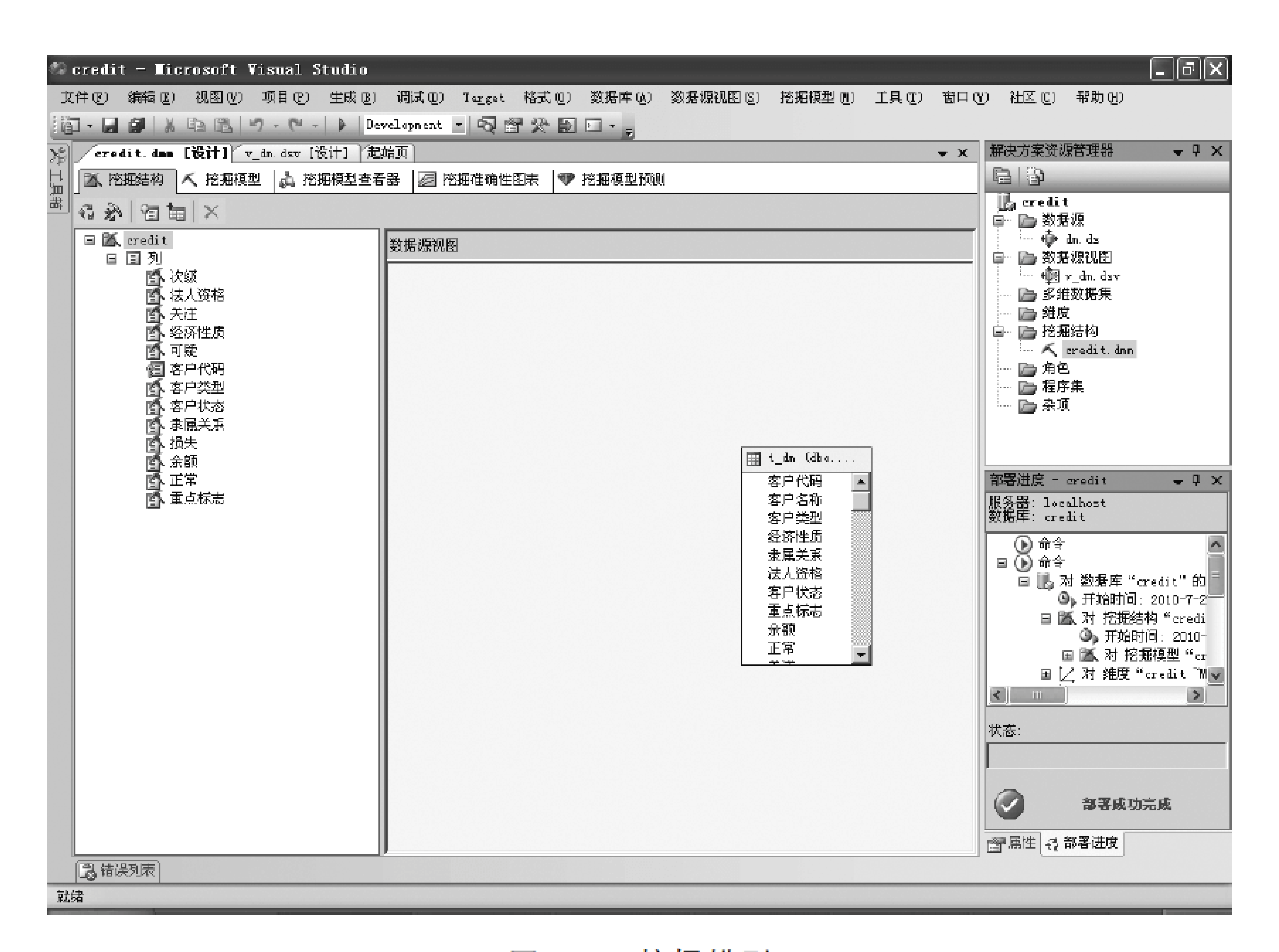


图 5-27　挖掘模型

可以在模型计算之前根据用户需要或者数据的特点对这些参数进行修改。在“挖掘模型”选项卡中，依照图 5-28 所示，右击 Microsoft Association Rules，从右键菜单选择“设置算法参数”，打开图 5-29 所示的参数设置对话框。

每个参数都已经设定了默认值和参数的可用范围，如果要使用不同的参数值，在参数所对应的列“值”中给出。表 5-5 对关联规则模型中设计到的 3 个主要指标进行简要说明。

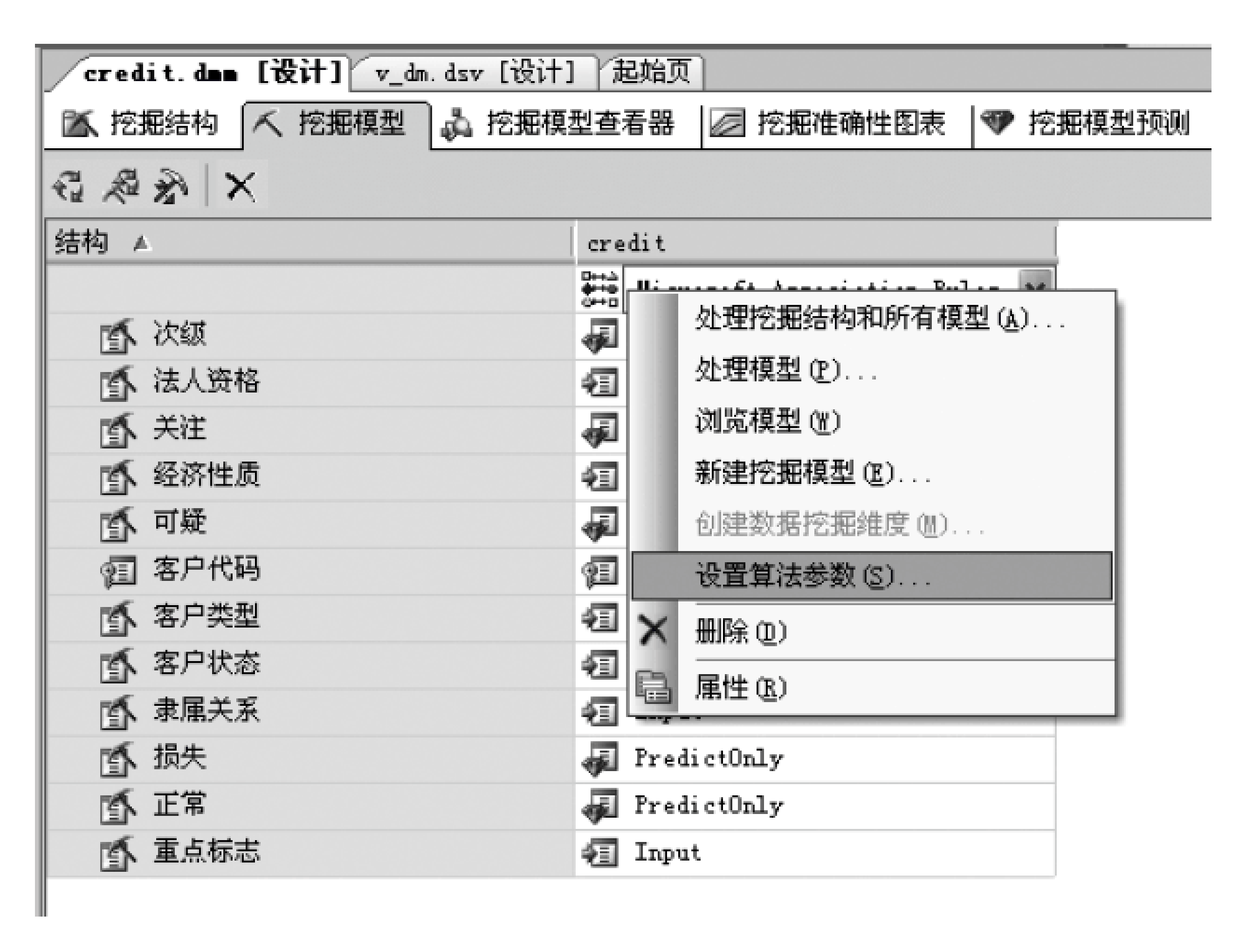


图 5-28　设置算法参数



图 5-29　参数设置

表 5-5　关键指标说明

指标名称	含　　义
支持度(Support)	支持度对项集形成有影响，用于描述项集出现频度。 最低支持度(Minimum_Support)意为只对达到指定频度的项集感兴趣，如果指定为小于 1 的值，则微软关联规则认为只对频度达到指定百分比的项集感兴趣。比如 0.03 表示项集支持度只有占到总项集数的 3%才能形成频繁项集。 最大支持度(Maximum_Support)则指定了项集出现频度的上限
概率(Probability)	概率对规则的形成有影响。 一条规则 A=>B 的概率定义为： Probability(A=>B)=Probability(B|A)=Support (A, B)/Support (A) 指定一定的最低概率值可以限制形成的规则数
重要性(Importance)	重要性对项集和规则形成均有影响。 定义为：Importance (A=>B)=log(p(B|A)/p(B|not A)) 如果该值为 0 表示 A 和 B 没有关联性，正值表示一旦拥有 A 则再拥有 B 的概率会增长，负值表示一旦拥有 A 则再拥有 B 的概率会降低

本例使用默认的参数。单击“取消”按钮回到主界面。

至此，就完成了数据挖掘模型的定制，在运行模型对所选数据进行分析之前，需要首先对项目进行部署。如图 5-30 所示，在资源管理器中用鼠标点击项目名称，打开右键菜单，选择“部署”功能。

图 5-31 显示项目部署进度。部署完成之后会在状态中显示完成提示。

部署成功之后，选择.dsv 页中的“挖掘结构”选项卡，单击挖掘结构名称，在右键菜单中选择“处理挖掘结构和所有模型”，如图 5-32。此功能是运行挖掘模型对数据进行分析，产生挖掘结果。



图 5-30　启动部署



图 5-31　显示部署进度

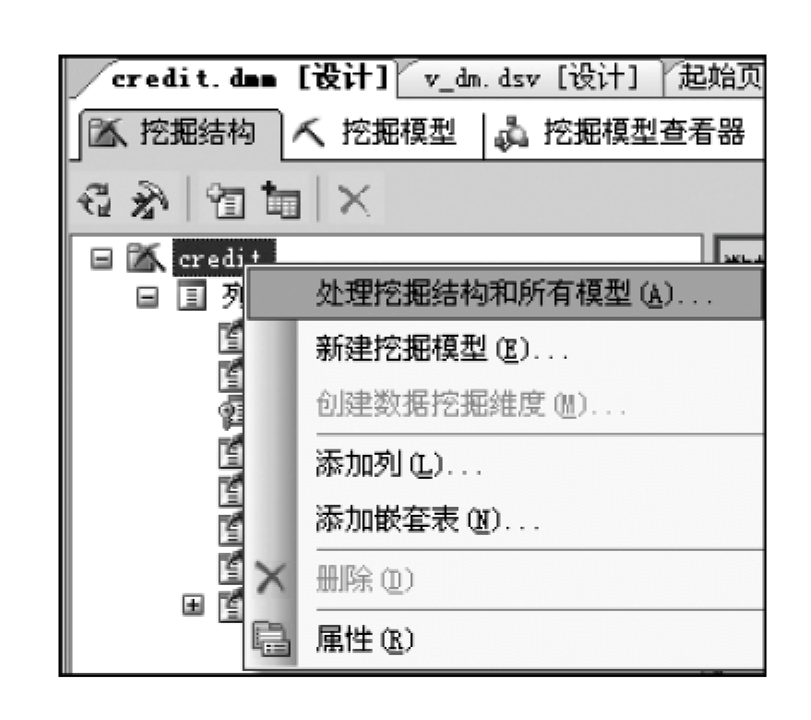


图 5-32　处理挖掘结构和所有模型

在图 5-33 中选择需要处理的挖掘结构，单击“运行”按钮，开始处理挖掘模型。



图 5-33　选择挖掘结构

处理完成，显示图 5-34 处理成功画面。



图 5-34 处理成功

3. 查看挖掘结果

模型处理完成之后，可以通过“挖掘模型查看器”查看挖掘结果。通过挖掘模型查看器，可以查看生成的项集、关联规则和项目之间的依赖关系。

第一次选择查看项目时，会显示一个加载模型的进度画面，如图 5-35 所示。所生成的项集数目、规则数目不同时，加载时间也有所不同。



图 5-35 加载挖掘模型进度

加载完成，就自动进入项集查看页面，图 5-36 是在默认条件下显示的项集，即最小支持度为 1，最小项集尺寸 0 的情况。也可以设置最小支持度和最小项集尺寸来减少显示的项集数量，如图 5-37 所示。

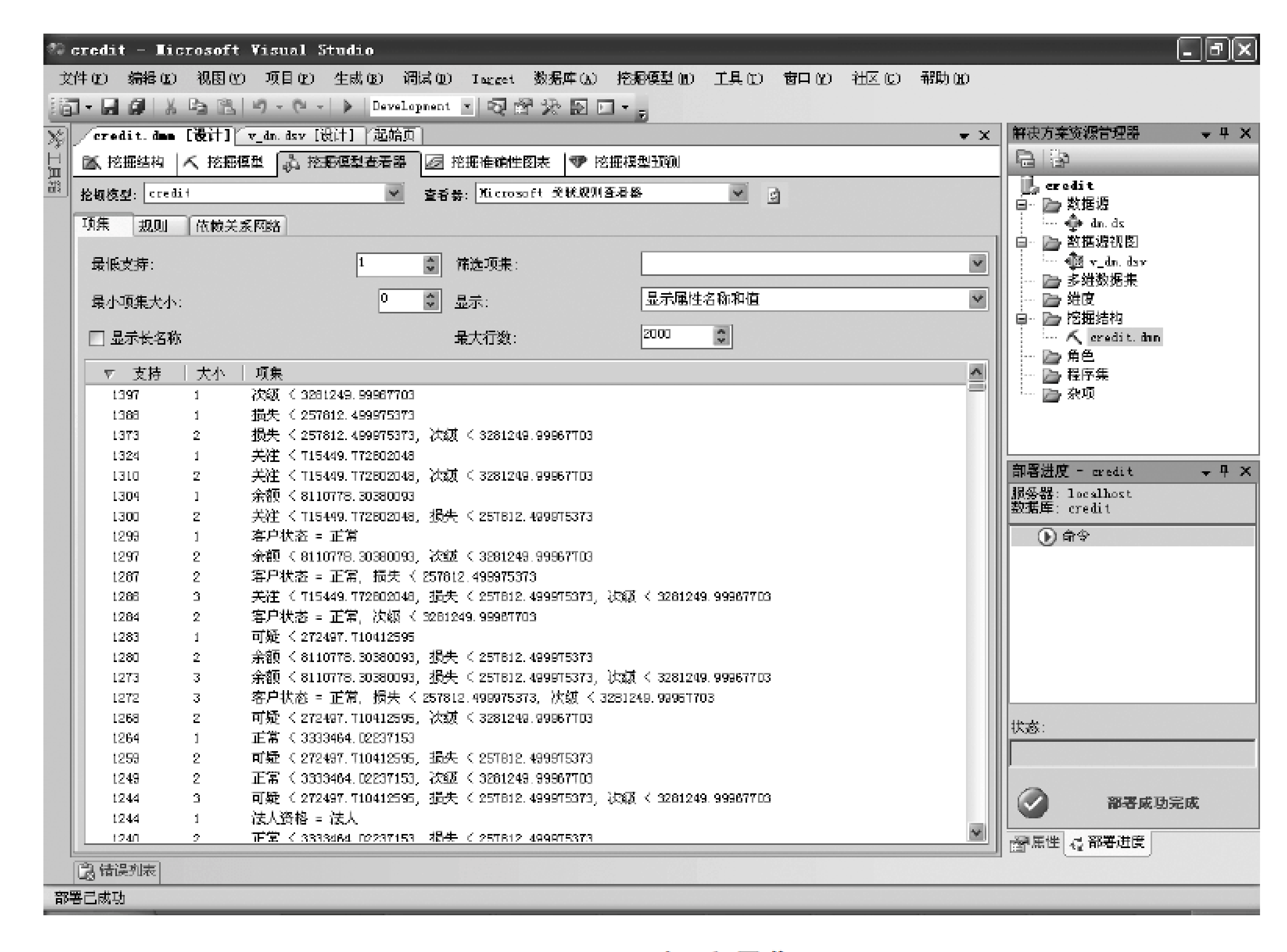


图 5-36 查看项集

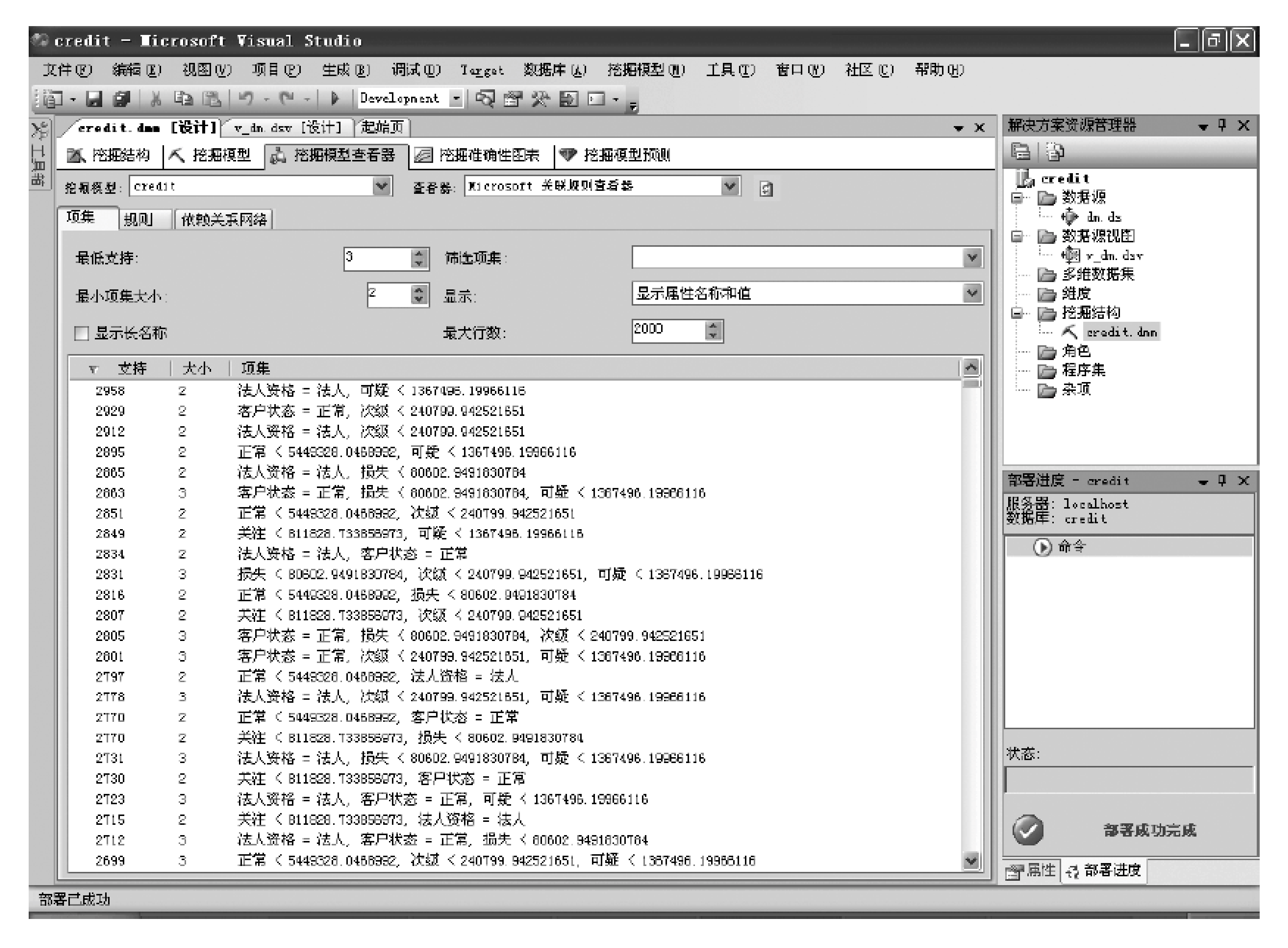


图 5-37　设定支持度和项集尺寸

在显示框上单击“支持”或“大小”，可以将项集按支持度大小或者长度排序。在“筛选项集”输入框中输入项目名称，可以只显示包含该项目的项集。“显示”框可以用来设定项集显示的形式，显示属性名称、显示属性值或者显示属性值和属性名称。“最大行数”设置在显示框中显示的项集数目。

“规则”选项卡用来查看产生的规则，如图 5-38 所示。同样也可以通过设定各种参数来筛选显示的规则。“最小概率”即最小置信度，其他选项功能同图 5-37。

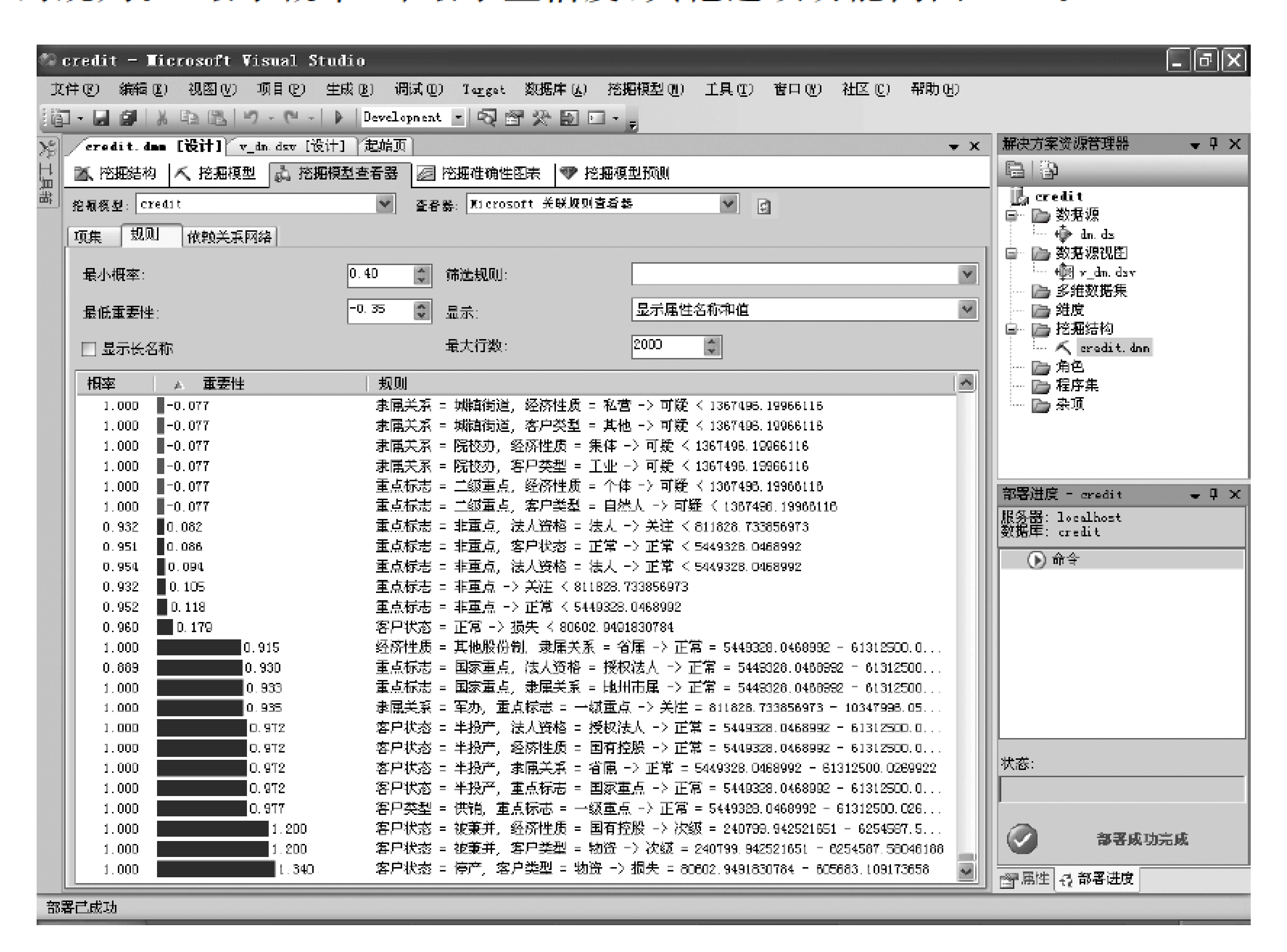


图 5-38　查看规则

图中规则的重要性与表 5-5 中所列的重要性意义相同，用蓝、红两种色条标记正、负两种值，用色条长度表示数值的大小。通过单击“重要性”列标题，可以将规则根据重要性排序，用户可以根据需要选择所关心的规则。

如其中“客户状态 = 停产，客户类型 = 物资 —> 损失 = 80602.9491830784 — 605683.109173658”的概率为 1，重要性1.34001993897064 就是一条规则。表示客户状态为“停产”、客户类型为“物资”的客户贷款业务中，银行损失在“80602.9491830784 — 605683.109173658”之间的可能性非常大。

“依赖关系网络”显示项目之间的依赖关系，如图 5-39 所示。当用鼠标选中一个结点，与其相关的结点将以特殊颜色显示；左边的调节按钮可以调节显示的依赖关系的强度；将鼠标指向一个结点，将会显示该结点的名称。

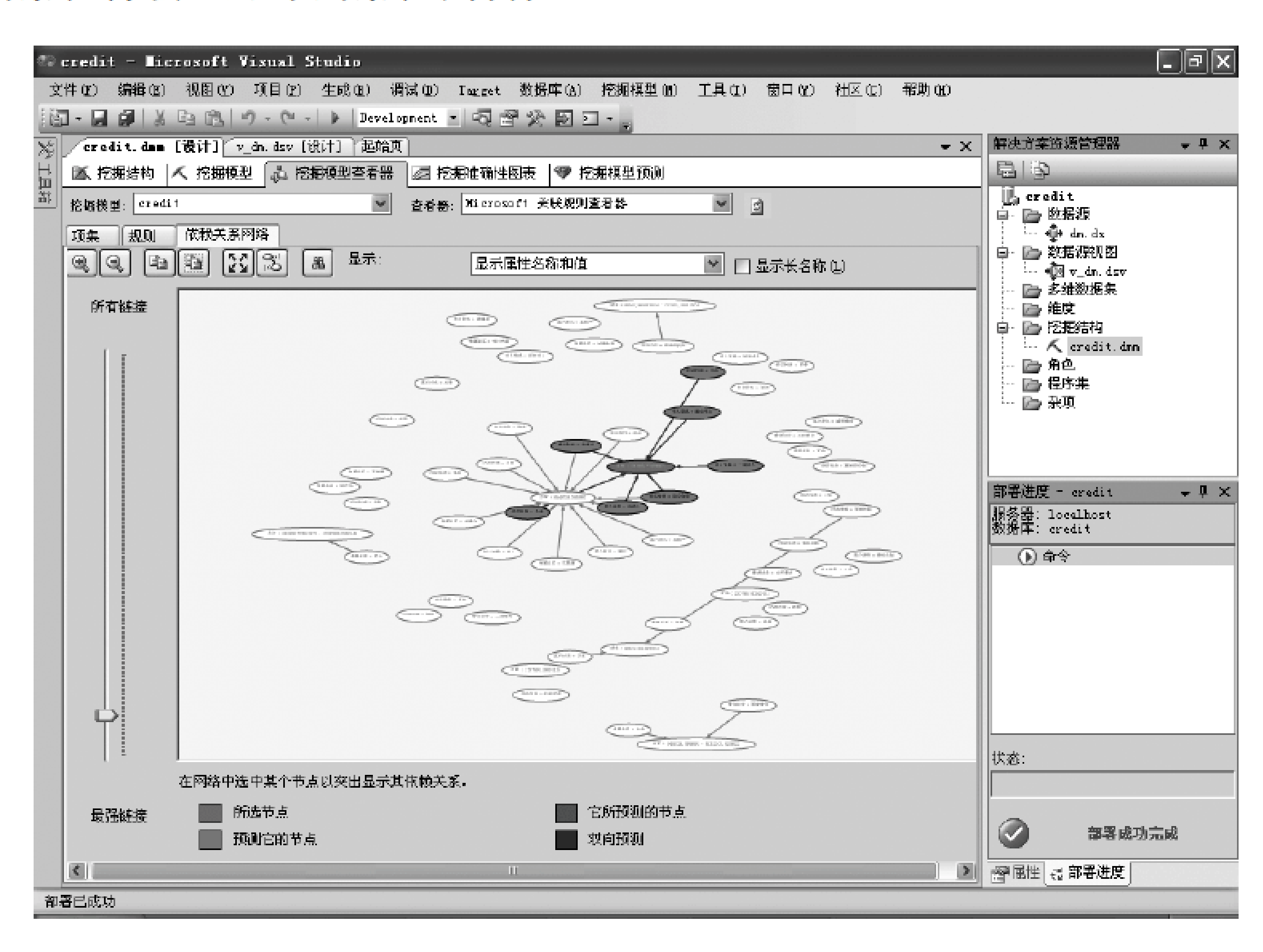


图 5-39　显示依赖关系

图 5-39 中的“查找结点”按钮（望远镜图标）可以帮助查找感兴趣的结点，当结点较多不容易找到某个结点时，在输入框中输入结点名称，会以特殊颜色显示包含该结点的依赖关系网络。单击“查找结点”按钮（望远镜图标）打开图 5-40 所示的对话框，从中选择感兴趣的结点，如“房地产开发”，确定之后回到关系网络（图 5-41），以“房地产开发”为当前结点的所有关系会用彩色显示。



图 5-40　查找感兴趣的结点

“挖掘准确性图表”以“提升图”或者“利润图”形式对挖掘结果的准确性给出描述。首先在图 5-42 所示的“列映射”选项卡中选择挖掘模型和输入表。

在实际应用中，这里选择的事例表应该与模型训练时所用的数据不同，是用来验证模型的准确性的。

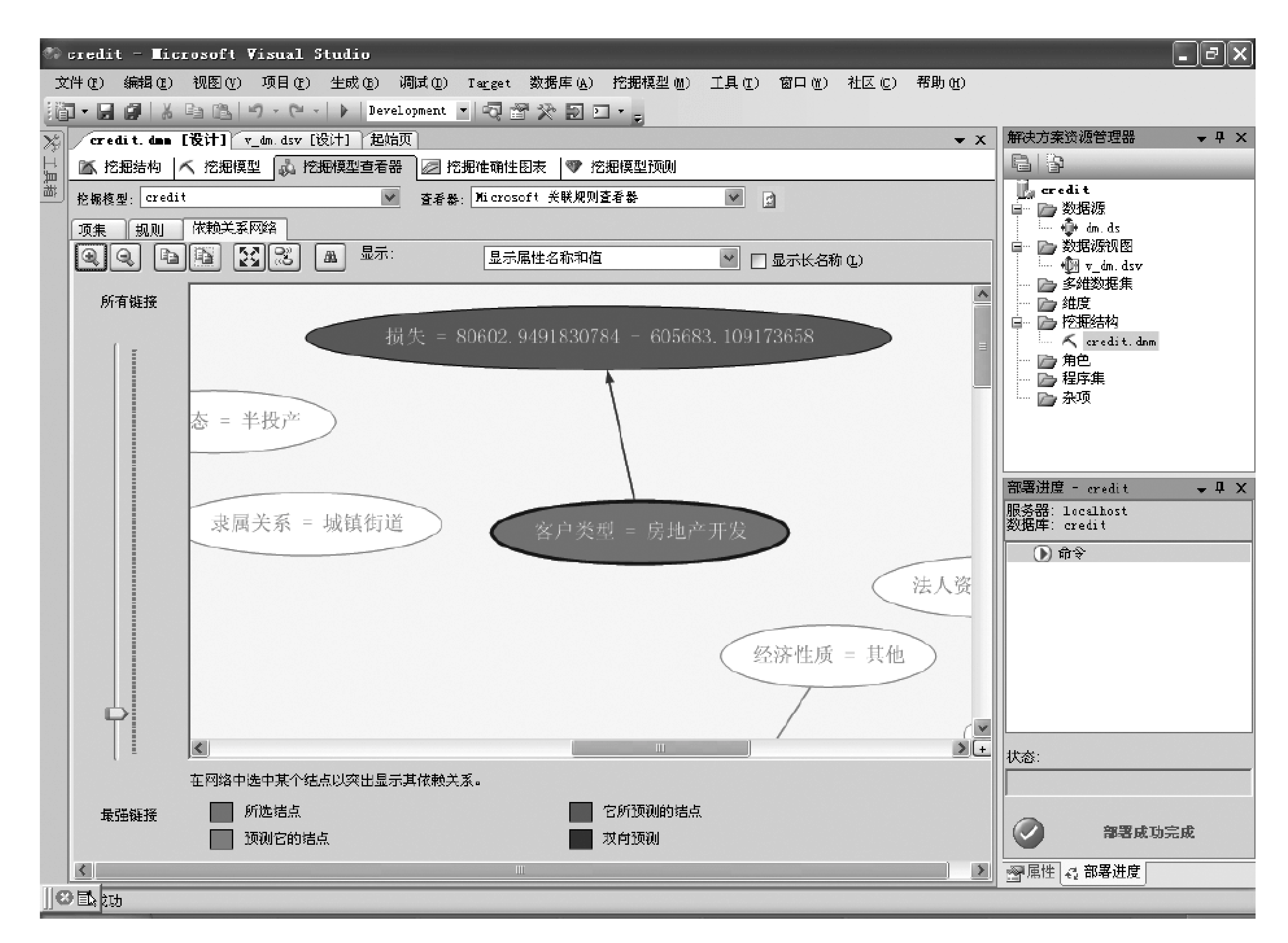


图 5-41 “房地产开发”的关系网络

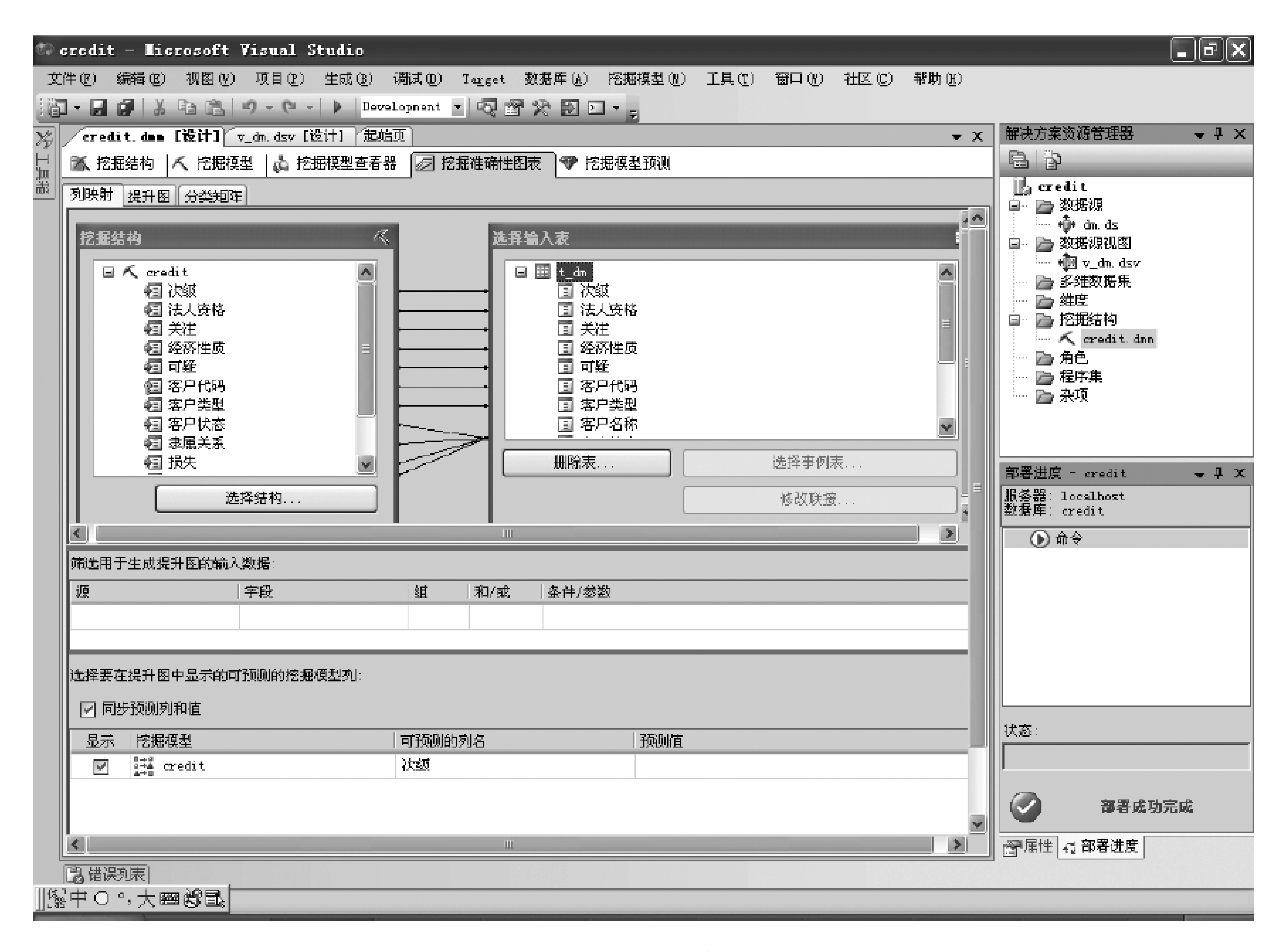


图 5-42 选择事例表

选定事例表之后,在界面下方选择可预测列,如图 5-43 所示。再进入“提升图”选项卡,将显示对选定的可预测列的预测准确性曲线,如图 5-44 所示。

可以通过选取“可预测的列名”,查看任意一个可预测列的预测准性图表。

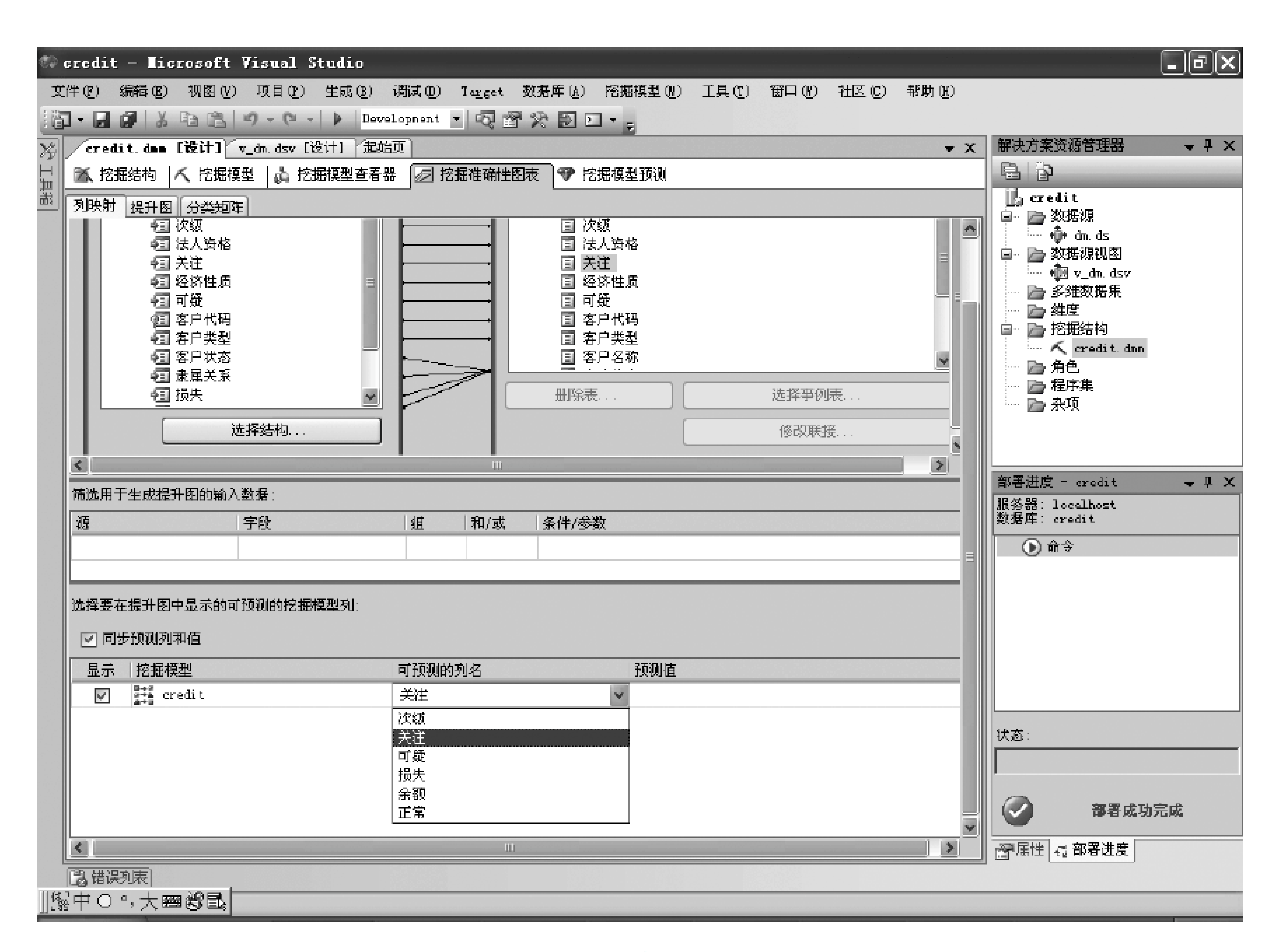


图 5-43　选择可预测列

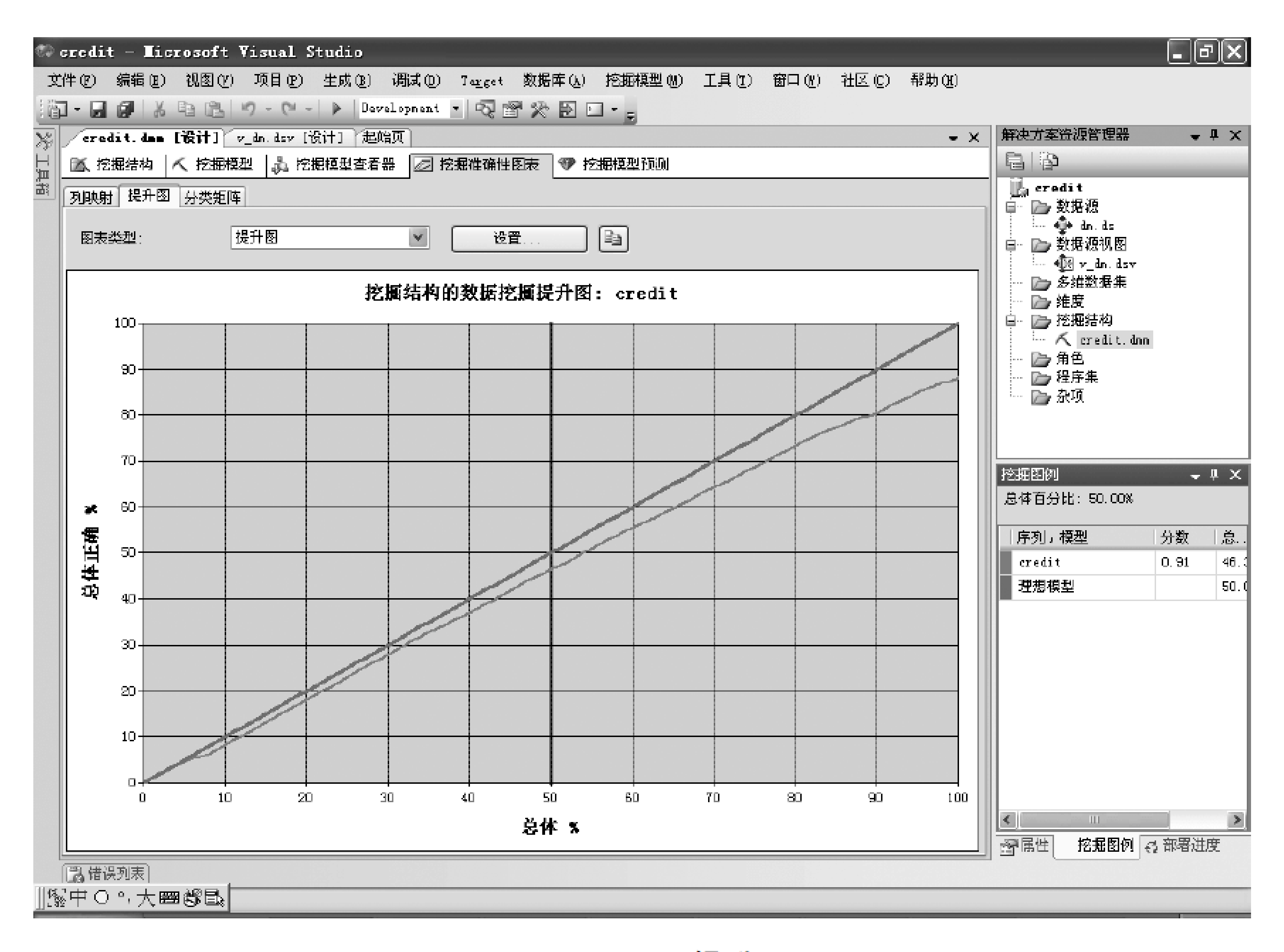


图 5-44　显示提升图

提升图中蓝色曲线表示理想模型，红色曲线表示模型 credit。右侧的挖掘图例中显示模型的分数，即准确度。

另一种显示模型准确性的工具是“分类矩阵”。与查看提升图一样，首先选择好可预测列，然后进入“分类矩阵”页面，将显示该列预测准确度的分类矩阵，如图 5-45 所示。

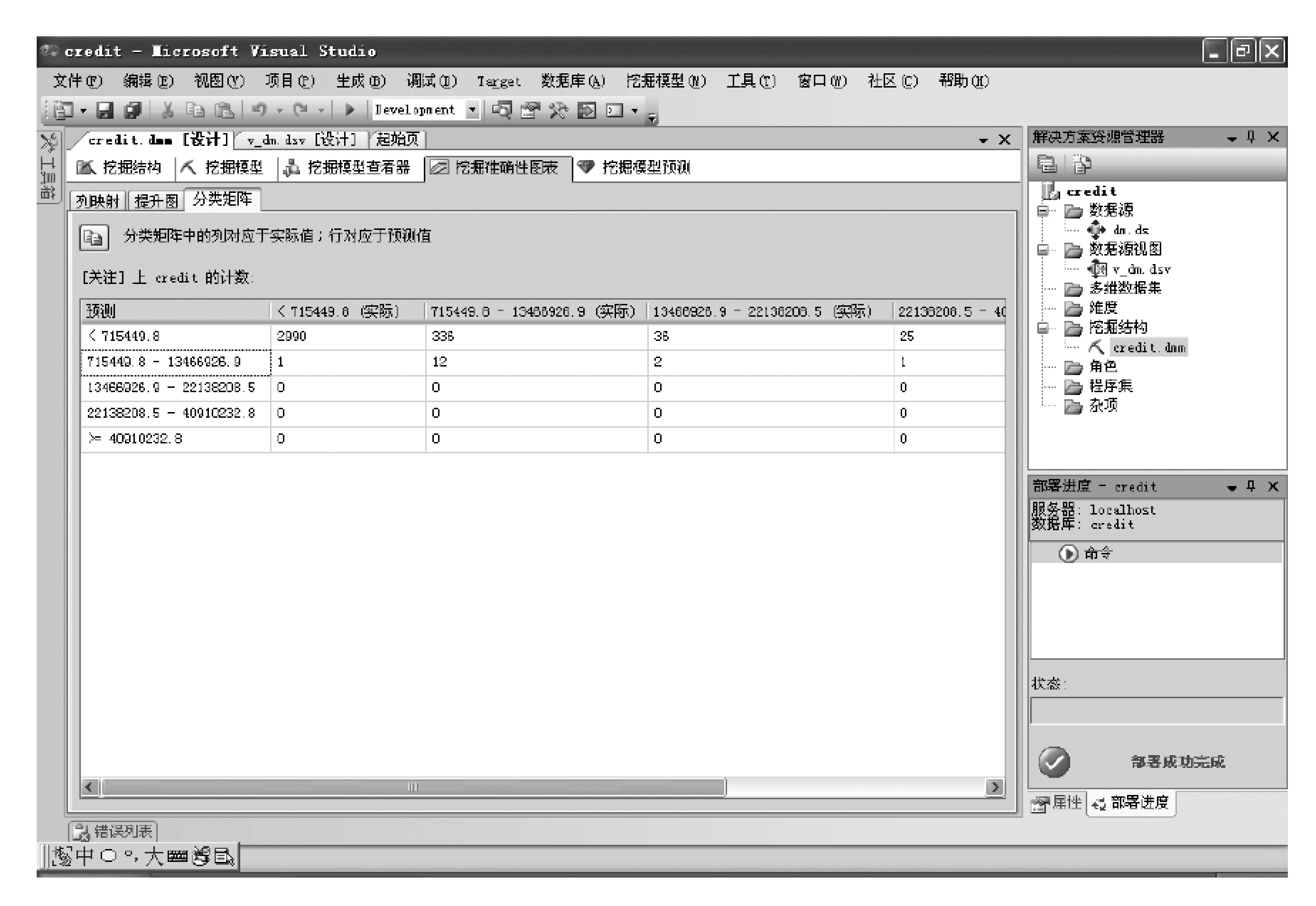


图 5-45　显示分类矩阵

分类矩阵中的列对应实际值，行对应预测值。如图中所示，列“关注”的值被分为“<715449.8”、“715449.8—13466926.9”、“13466926.9—22038208.5”等 5 个段，这些划分是模型在训练过程中自动实现的，无须用户干预。行列交叉处的单元格值表示满足该实际值和预测值的记录数目。如实际值<715449.8，预测值<715449.8 的记录数目为 2990；实际值<715449.8，预测值∈(715449.8—13466926.9]的记录数目为 1，通过观察分类矩阵，可以发现模型对该字段的预测准确度。

模型产生之后，用户可以对产生的规则进行筛选，通过针对性的市场调查、专家评估等方式确定哪些规则是真正可用的，以便在经营决策中使用。

小结

本章介绍了数据挖掘中的关联规则挖掘方法，包括关联规则的概念、起源，介绍了关联规则挖掘的两个子问题——发现频繁项集和产生规则。重点介绍了发现频繁项集这一任务中，产生候选项集和不产生候选项集两类重要方法。详细描述了 Apriori 算法和 FP-Growth 算法的原理和计算过程，给出了算法的伪代码描述，并且通过实例演示了算法的计算过程。最后应用银行贷款业务数据库，用 SQL Server 2005 提供的数据挖掘工具进关联规则挖掘，文中介绍了 SQL Server 2005 的 Analysis Services 的操作方法和整个挖掘过程。

当挖掘问题是希望根据事物的某些属性预测可能发生的情况，即发现形如 A⇒B 的知识时，可以采用关联规则挖掘方法。

习题 5

1．说明关联规则挖掘的目的和作用。

2．简要说明在频繁模式发现技术中，产生候选项集和不产生候选项集两种技术各自的特点和优缺点。

3．图 5-1 所示的数据库，如果分别设定最小支持度 $s=10\%$ 和 $s=40\%$，置信度 $c=70\%$，计算该示例数据库中的频繁项集和规则。

4．根据图 5-3 所示的 FP-tree，找出以 I4，I3，I1，I2 为后缀的频繁模式。

5．根据图 5-3 所示的 FP-tree，找出其中所有的关联规则及他们的置信度。

6．练习使用 SQL Server 2005 的关联规则挖掘模型。

第 6 章　决策树方法

决策树(decision tree)方法(也称为判定树算法)是最受欢迎的数据挖掘技术之一,主要用于分类和预测。决策树学习是以样本为基础的归纳学习方法,将决策树转换成分类规则比较容易。决策树的表现形式类似于流程图的树结构,在决策树的内部结点进行属性测试,并根据属性值判断由该结点引出的分支,在叶结点得到结论。内部结点是属性或属性的集合,叶结点代表样本所属的类或类分布。基于决策树的学习算法在学习过程中不需要用户了解很多背景知识,只要训练样本能够用“属性-值”的方式表述,就可以使用该算法来学习。

决策树学习的基本算法是贪心算法,采用自顶向下的递归方式构造决策树。Hunt 等人于 1966 年提出的概念学习系统(conception learning system,CLS)是最早的决策树算法,以后的许多决策树算法都是对 CLS 算法的改进或由 CLS 衍生而来。

澳大利亚悉尼大学的 Ross Quinlan 于 1979 年提出了著名的 ID3(information decision)方法。1993 年,Quinlan 提出了以 ID3 为蓝本的 C4.5 算法,可以处理数值属性、缺失值和噪声数据,是一个能处理连续属性的算法。其他决策树方法还有 ID3 的增量版本 ID4 和 ID5 等。本章主要介绍 ID3 和 C4.5 算法。

6.1　信息论的基本原理

决策树方法是基于信息论的数据挖掘方法中的一种,为便于理解,下面先来介绍信息论的基本原理。

6.1.1　信息论原理

信息论是 C. E. Shannon 为解决信息传递(通信)过程问题而建立的理论,也称为统计通信理论。一个传递信息的系统是由发送端(信源)、接收端(信宿)以及连接两者的通道(信道)三者组成。信息论把通信过程看作是在随机干扰的环境中传递信息的过程。在这个通信模型中,信源和干扰(噪声)都被理解为某种随机过程或随机序列。因此,在进行实际的通信之前,信宿不可能确切了解信源究竟会发出什么样的具体信息,不可能判断信源会处于什么样的状态,这种情形就称为信宿对于信源状态具有不确定性,而且这种不确定性是存在于通信之前的,因而又叫做先验不确定性。

在进行通信之后,信宿收到了信源发来的信息,这种先验不确定性才会被消除或者被减少。如果干扰很少,不会对传递的信息产生任何可察觉的影响,信源发出的信息能够被信宿全部收到,在这种情况下,信宿的先验不确定性就会被完全消除。但是,在一般情况下,干扰总会对信源发出的信息造成某种破坏,使信宿收到的信息不完全。因此,先验不确定性不能全部被消除,只能部分地消除。换句话说,通信结束之后,信宿仍然具有一定程度的不确定性,这就是后验不确定性。显然,后验不确定性总要小于先验不确定性,不可能大于先验不确定性。

如果后验不确定性的大小刚好等于先验不确定性的大小，这就表示信宿根本没有收到信息；如果后验不确定性等于零，这就表示信宿收到了全部信息。可见信息是用来消除不确定性（随机）的度量，信息量的大小由所消除的不确定性的大小来衡量。

6.1.2 互信息的计算

1. 定义

（1）设 S 为训练集，训练集中每个训练样本有 n 个特征（属性），表示为 $(A_1, A_2, \cdots, A_n)$，$|S|$ 表示样本总数。

（2）S 中有 U_1, U_2 两类。$|U_i|$ 表示 U_i 类例子数。

（3）特征 A_k 处有 m 个取值，分别为 $(V_1, V_2, \cdots, V_m)$。

2. 出现概率

U_i 类出现概率为

$$P(U_i) = |U_i| / |S| \tag{6-1}$$

自然有

$$\sum_{i=1}^{2} P(U_i) = 1 \tag{6-2}$$

3. 条件概率

U_i 类中在特征 A_k 处，取值 V_j 的样本集合 V_{ij} 的条件概率为

$$P(V_j \mid U_i) = |V_{ij}| / |U_i| \tag{6-3}$$

自然有

$$\sum_{i=1}^{m} P(V_j \mid U_i) = 1 \tag{6-4}$$

4. 子集概率

在特征 A_k 处，取值 V_j 的样本集合的概率为

$$P(V_j) = |V_j| / |S| \tag{6-5}$$

自然有

$$\sum_{i=1}^{m} P(V_j) = 1 \tag{6-6}$$

5. 子集条件概率

在特征 A_k 处取值 V_j 的例子，属于 U_i 类的例子集合 U_{ij} 的条件概率为

$$P(U_i \mid V_j) = |U_{ij}| / |V_j| \tag{6-7}$$

自然有

$$\sum_{i=1}^{2} P(U_i \mid V_j) = 1 \tag{6-8}$$

6. 信息熵

（1）消息传递系统由消息的信源、信宿以及信道组成。

（2）消息（符号）$U_i(i=1,2,\cdots,q)$ 的发生概率 $P(U_i)$ 组成信源数学模型（样本空间或概率空间），即

$$[U, P] = \begin{bmatrix} U_1 & U_2 & \cdots & U_q \\ P(U_1) & P(U_2) & \cdots & P(U_q) \end{bmatrix} \tag{6-9}$$

(3) 自信息。消息 U_i 发生后所含有的信息量。它反映了消息 U_i 发生前的不确定性，定义为

$$I(U_i) = \log_k \frac{1}{P(U_i)} = -\log_k P(U_i) \tag{6-10}$$

当 $k=2$ 时，所得的信息量单位为 bit(位)。

(4) 信息熵。信息熵是信源输出后，每个消息所提供的平均信息量，也反映了信源输出前的平均确定性。定义为

$$H(U) = \sum_i P(U_i)\log_k \frac{1}{P(U_i)} = -\sum_i P(U_i)\log_k P(U_i) \tag{6-11}$$

信息熵 $H(U)$是信源输出前的平均不确定性，也称为先验熵。其性质如下：

① $H(U)=0$ 时，说明只存在着唯一的可能性，不存在不确定性。

如果 n 种可能的发生都有相同的概率，即所有的 U_i 有 $P(U_i)=1/n$，$H(U)$达到最大值 $\log n$，系统的不确定性最大。

② $P(U_i)$互相接近，$H(U)$就大；$P(U_i)$相差大，$H(U)$就小。

如果信道中无干扰，信道输出符号与输入符号一一对应，那么接收到传送过来的符号后就消除了对发送符号的先验不确定性。

7. 互信息

(1) 后验熵和条件熵。一般信道中有干扰存在，接收到符号 V 后对发送的是什么符号仍有不确定性。那么，怎样来度量接收到 V 后，关于 U 的不确定性呢？当没有接收到输出符号 V 时，已知输入符号 U 的概率分布为 $P(U)$，而当接收到输出符号 $V=V_j$ 后，输入符号的概率分布发生了变化，变成后验概率分布 $P(U|V_j)$。那么接收到输出符号 $V=V_j$ 后，关于 U 的平均不确定性为

$$H(U \mid V_j) = \sum_j P(U_i \mid V_j)\log \frac{1}{P(U_i \mid V_j)} \tag{6-12}$$

这是接收到输出符号 $V=V_j$ 后关于 U 的后验熵。后验熵是当信道接收端接收到输出符号 $V=V_j$ 后，关于输入符号 U 的信息度量，即

$$H(U \mid V) = \sum_j P(V_j)\sum_j P(U_i \mid V_j)\log \frac{1}{P(U_i \mid V_j)} \tag{6-13}$$

后验熵在输出符号集 V 的范围内是个随机量，对后验熵在输出符号集 V 中求期望，得到条件熵。

这个条件熵称为信道疑义度。它表示在信宿收到全部输出符号 V 后，对于信源的符号集 U 尚存在不确定性(疑义)，对 U 集尚存在的不确定性是由干扰引起的。如果是一一对应信道，那么接到符号集 V 后，对 U 集的不确定性完全消除，则信道疑义度 $H(U|V)=0$。

从上面分析可知：条件熵小于无条件熵，即 $H(U|V)<H(U)$。说明收到符号集 V 的所有符号后，关于输入符号 U 的平均不确定性减小了。即总能消除一些关于输入端 U 的不确定性，从而获得了一些信息。

(2) 平均互信息。$H(U)$代表接收到符号集 V 以前关于输入符号集 U 的平均不确定性，而 $H(U|V)$代表收到符号集 V 后关于输入符号 U 的平均不确定性。可见，通过信道传输消除了一些不确定性，获得了一定的信息。定义为

$$I(U,V) = H(U) - H(U \mid V) \tag{6-14}$$

$I(U,V)$称为 U 和 V 之间的平均互信息，它代表接收到符号集 V 后获得的关于 U 的信息量。

可见，熵 $H(U)$、$H(U|V)$只是平均不确定性的描述。熵差 $H(U)-H(U|V)$是不确定性的消除，即互信息才是接收端所获得的信息。

6.2 常用决策树算法

6.2.1 ID3 算法

1. 基本思想

在实体世界中，每个实体用多个特征来描述。每个特征限于在一个离散集中取互斥的值。例如，设实体是某天早晨，分类任务是判断是否适合打网球。实体的特征由 4 个属性标识：

天气：取值为晴、多云、雨；

气温：取值为冷、适中、热；

湿度：取值为高、正常；

风力：取值为有风、无风。

实体(即某天早晨)特征描述为：多云，冷，湿度正常，无风。

它是否适合打网球呢？要解决这个问题，需要用某个规则来判定，这个规则来自于大量的实际例子，从例子中总结出规则，有了规则就可以对任何实体做出判定。

每个实体在现实世界中属于不同的类别。为简单起见，假定有两个类别 P 和 N，分别代表“适合”和“不适合”，在这两个类别的归纳任务中，P 类和 N 类的实体分别称为概念的正例和反例。将一些已知的正例和反例放在一起便得到训练集。

表 6-1 给出一个训练集。由 ID3 算法可得出一棵正确分类训练集中每个实体的决策树，如图 6-1 所示。

表 6-1　ID3 算法训练样本集

编号	属　性				类别
	天气	气温	湿度	风力	
1	晴	热	高	无风	N
2	晴	热	适中	无风	N
3	多云	热	高	无风	P
4	雨	适中	高	无风	P
5	雨	冷	正常	无风	P
6	雨	冷	正常	有风	N
7	多云	冷	正常	有风	P
8	晴	适中	高	无风	N
9	晴	冷	正常	无风	P
10	雨	适中	正常	无风	P
11	晴	适中	正常	有风	P
12	多云	适中	高	有风	P
13	多云	热	正常	无风	P
14	雨	适中	高	有风	N

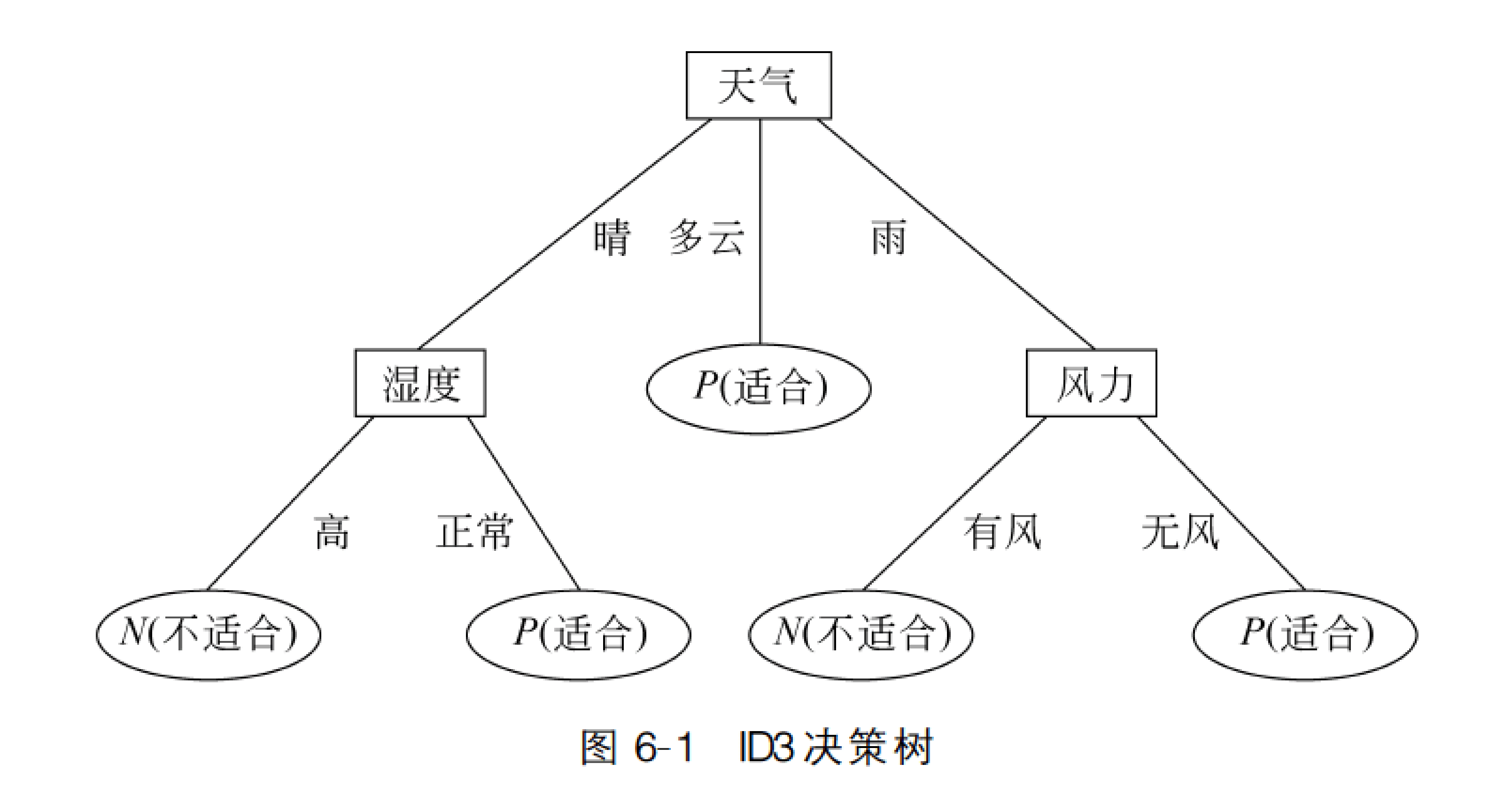


图 6-1　ID3 决策树

决策树叶子为类别名，即 P 或 N。其他结点由实体的特征组成，每个特征的不同取值对应不同分支。若要对一个实体分类，从树根开始进行测试，按特征的取值分支向下进入下层结点，对该结点进行测试，过程一直进行到叶结点，实体被判为属于该叶结点所标记的类别。现用图 6-1 来判定本节开始处的例子，得该实体的类别为 P 类。ID3 就是要从表 6-1 的训练集构造图 6-1 这样的决策树。

实际上，能正确分类训练集的决策树不只一棵，Quinlan 的 ID3 算法能得出结点最少的决策树。

2. 主算法

(1) 从训练集中随机选择一个既含正例（图 6-1 中类别 P）又含反例（图 6-1 中类别 N）的子集（称为“窗口”）；

(2) 用“建树算法”对当前窗口形成一棵决策树；

(3) 对训练集（窗口除外）中例子用所得决策树进行类别判定，找出错判的例子；

(4) 若存在错判的例子，把它们插入窗口，转(2)，否则结束。

主算法流程如图 6-2 所示。其中 PE、NE 分别表示正例集和反例集，它们共同组成训练集。PE′、PE″和 NE′、NE″分别表示正例集和反例集的子集。

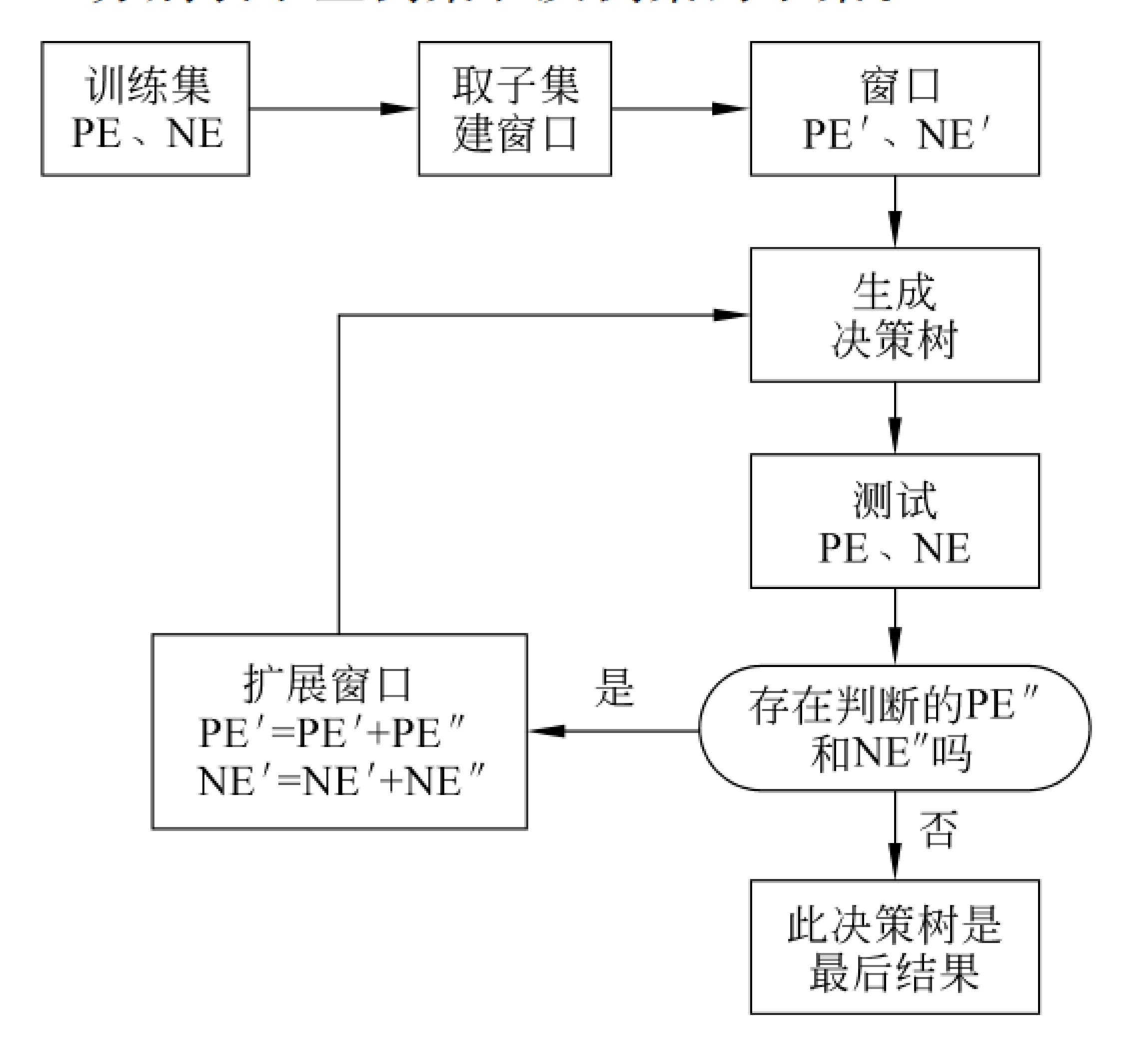


图 6-2　ID3 算法流程

主算法中每迭代循环一次，生成的决策树将会不相同。

3. 建树算法

(1) 对当前例子集合，计算各特征的互信息；

(2) 选择互信息最大的特征 A_k；

(3) 把在 A_k 处取值相同的例子归于同一子集，A_k 取几个值就得几个子集；

(4) 对既含正例又含反例的子集，递归调用建树算法；

(5) 若子集仅含正例或反例，对应分枝上标 P 或 N，返回调用处。

4. 实例计算

对于表 6-1 所示的分类问题，采用 ID3 算法构建决策树的方法如下。

(1) 信息熵的计算。信息熵的计算公式参见式(6-5)和式(6-11)，对于 9 个正例和 5 个反例，有

$$P(U_1) = 9/14, \quad P(U_2) = 5/14$$
$$H(U) = (9/14)\mathrm{lb}(14/9) + (5/14)\mathrm{lb}(14/5) = 0.94$$

(2) 条件熵计算。条件熵的计算公式参见式(6-7)和式(6-13)。

当 A_1 = 天气时，取值 V_1 = 晴，V_2 = 多云，V_3 = 雨。在 A_1 处取值“晴”的例子 5 个，取值“多云”的例子 4 个，取值“雨”的例子 5 个，故

$$P(V_1) = 5/14, \quad P(V_2) = 4/14, \quad P(V_3) = 5/14$$

取值为“晴”的 5 个例子中有 2 个正例，3 个反例，故

$$P(U_1 \mid V_1) = 2/5, \quad P(U_2 \mid V_1) = 3/5$$

同理有

$$P(U_1 \mid V_2) = 4/4, \quad P(U_2 \mid V_2) = 0$$
$$P(U_1 \mid V_3) = 2/5, \quad P(U_2 \mid V_3) = 3/5$$
$$\begin{aligned} H(U \mid V) =& (5/14)((2/5)\mathrm{lb}(5/2) + (3/5)\mathrm{lb}(5/3)) \\ &+ (4/14)((4/4\mathrm{lb}(4/4) + 0) \\ &+ (5/14)((2/5)\mathrm{lb}(5/2) + 3/5\mathrm{lb}(5/3)) = 0.694 \end{aligned}$$

(3) 互信息计算。当对 A_1 = 天气时，有

$$I(\text{天气}) = H(U) - H(U \mid V) = 0.94 - 0.694 = 0.246$$

类似可得

$$I(\text{气温}) = 0.029$$
$$I(\text{湿度}) = 0.151$$
$$I(\text{风力}) = 0.048$$

(4) 建决策树的树根和分支。ID3 算法将选择互信息最大的特征“天气”作为树根，对 14 个样本根据“天气”的 3 个取值进行划分，3 个分支对应 3 个子集，分别是

$$F_1 = \{1,2,8,9,11\}, \quad F_2 = \{3,7,12,13\}, \quad F_3 = \{4,5,6,10,14\}$$

其中，F_2 中的例子全部属于 P 类，因此对应分枝标志为 P，其余两个子集既含有正例又含有反例，将递归调用建树算法。

(5) 递归建树。分别对 F_1 和 F_3 子集利用 ID3 算法，在每个子集中对各特征(仍为 4 个特征)求互信息。

① F_1 中的天气全取“晴”值，则 $H(U) = H(U|V)$，有 $I(U|V) = 0$，在余下 3 个特征中求

出“湿度”互信息最大，以它为该分支的根结点，再向下分支。“湿度”取“高”的例子全为 N 类，该分支标记为 N；取值“正常”的例子全为 P 类，该分支标记为 P。

② 在 F_3 中，对 4 个特征求互信息，得到“风力”特征互信息最大，则以它为该分支根结点，再向下分支，“风力”取“有风”时全为 N 类，该分支标记为 N；取“无风”时全为 P 类，该分支标记为 P。

这样就得到了图 6-1 的决策树。

6.2.2 C4.5 算法

C4.5 算法是从 ID3 算法演变而来，除了拥有 ID3 算法的功能外，C4.5 算法引入了新的方法并增加了新的功能。例如用信息增益比例的概念；合并具有联系属性的值；可以处理缺少属性值的训练样本；k 交叉验证；规则的产生方式等。

1. 信息增益比例的概念

信息增益比例是在信息概念的基础上发展起来的，一个属性的信息增益的比例用如下公式给出。

$$\mathrm{GainRation}(A) = \frac{\mathrm{Gain}(A)}{\mathrm{SplitI}(A)} \tag{6-15}$$

其中，

$$\mathrm{SplitI}(A) = -\sum_{j=1}^{v} p_j \mathrm{lb}(p_j) \tag{6-16}$$

这里设置属性 A 具有 v 个不同值 $\{a_1, a_2, \cdots, a_v\}$。可以用属性 A 将 S 划分为 v 个子集 $\{S_1, S_2, \cdots, S_v\}$，其中 S_j 包含 S 中这样一些样本：它们在 A 上具有值 a_j。假如以属性 A 的值为基准，对样本进行分割，SplitI(A)就是前面熵的概念。

2. 连续属性值的处理

ID3 算法最初假定属性值是离散的，但在实际环境中，很多属性是连续型的。对于连续属性值，C4.5 处理过程如下：

(1) 根据属性值对数据集排列；

(2) 用不同的阈值将数据集动态地进行划分；

(3) 当输出改变时确定一个阈值；

(4) 取两个实际值中的中点作为一个阈值；

(5) 取两个划分，所有样本都在这两个划分中；

(6) 得到所有可能的阈值、增益及增益比；

(7) 每一个属性会变为两个取值，即小于阈值或大于等于阈值。

针对属性为连续数值的情况，比如属性 A 有连续的属性值，则在训练集中可以按升序方式排列 $a_1, a_2, \cdots, a_m$ (m 为训练样本的个数)。如果 A 共有 n 种取值，则对每个取值 $v_j (j=1,2,\cdots,n)$ 将所有的记录进行划分。这些记录被划分成两个部分：一部分落入在 v_j 的范围内，而另一部分则大于 v_j。针对每个划分分别计算增益比率，选择增益最大的划分来对相应的属性进行离散化。

3. 未知属性值的处理

C4.5 处理的样本中可以含有未知属性值，其处理方法是用最常用的值替代，或者是将

最常用的值分在同一类中。具体采用概率的方法,依据属性已知的值,对属性的每一个取值赋予一个概率。

4. 规则的产生

一旦树建立,就可以把树转化成 if-then 规则。规则存储于一个二维数组中,每一行代表树中的一个规则,即从根到叶之间的一个路径,表中的每列存放着树中的一个结点。

5. 案例计算

下面以实际的例子,详细地说明 C4.5 分类实现的过程。所采用的数据集如表 6-2 所示,包含 5 个属性:天气(离散属性),气温(离散属性),湿度(连续属性),风力(离散属性),是否适合打网球(类别属性)。

表 6-2　C4.5 算法训练样本集

编号	属性				类别
	天气	气温	湿度	风	
1	晴	热	85	无风	N
2	晴	热	90	无风	N
3	多云	热	78	无风	P
4	雨	适中	96	无风	P
5	雨	冷	80	无风	P
6	雨	冷	70	有风	N
7	多云	冷	65	有风	P
8	晴	适中	95	无风	N
9	晴	冷	70	无风	P
10	雨	适中	80	无风	P
11	晴	适中	70	有风	P
12	多云	适中	90	有风	P
13	多云	热	75	无风	P
14	雨	适中	80	有风	N

首先对属性"湿度"进行离散化,针对上面的训练集合,通过检测每个划分确定最好的划分在 75 处,因为在 75 处落在两边的个数基本相等,则这个属性的范围就变为{≤75,>75}。

然后计算类别属性分类的信息熵:

类别中出现 9 个 P 类和 5 个 N 类,因此有

$$P(S_1)=9/14,\quad P(S_2)=5/14$$

$$I(s_1,s_2)=(9/14)\mathrm{lb}(14/9)+(5/14)\mathrm{lb}(14/5)=0.940$$

同理计算"天气"属性的 SplitI 值得到

$$\mathrm{SplitI}(天气)=-(5/14)\mathrm{lb}(5/14)-(4/14)\mathrm{lb}(4/14)-(5/14)\mathrm{lb}(5/14)=1.577$$

对于决策类别,天气属性每个取值的条件熵为

(1) 天气取值为"晴",$s_{11}=2$,$s_{21}=3$,$I(s_{11},s_{21})=0.9707$。

(2) 天气取值为"多云",$s_{12}=4$,$s_{22}=0$,$I(s_{12},s_{22})=0$。

(3) 天气取值为"雨",$s_{13}=3$,$s_{23}=2$,$I(s_{13},s_{23})=0.9707$。

因此,得到天气属性的条件熵为

$$H(类别 \mid 天气) = \frac{5}{14} \times 0.9707 + 0 + \frac{5}{14} \times 0.9707 = 0.6933$$

对应的信息增益为

$$\text{Gain}(天气) = H(类别) - H(类别 \mid 天气) = 0.940 - 0.6933 = 0.2467$$

最后得到互信息为

$$\text{GainRatio}(天气) = \frac{0.2467}{1.577} = 0.156$$

同理,可以计算出

$$\text{GainRatio}(风力) = 0.049$$

$$\text{GainRatio}(温度) = 0.0248$$

$$\text{GainRatio}(湿度) = 0.0483$$

选出最大的 GainRatio(天气)=0.156。根据天气的取值,可以得到 3 个分支,同时数据集被划分成 3 个子集,如图 6-3 所示。

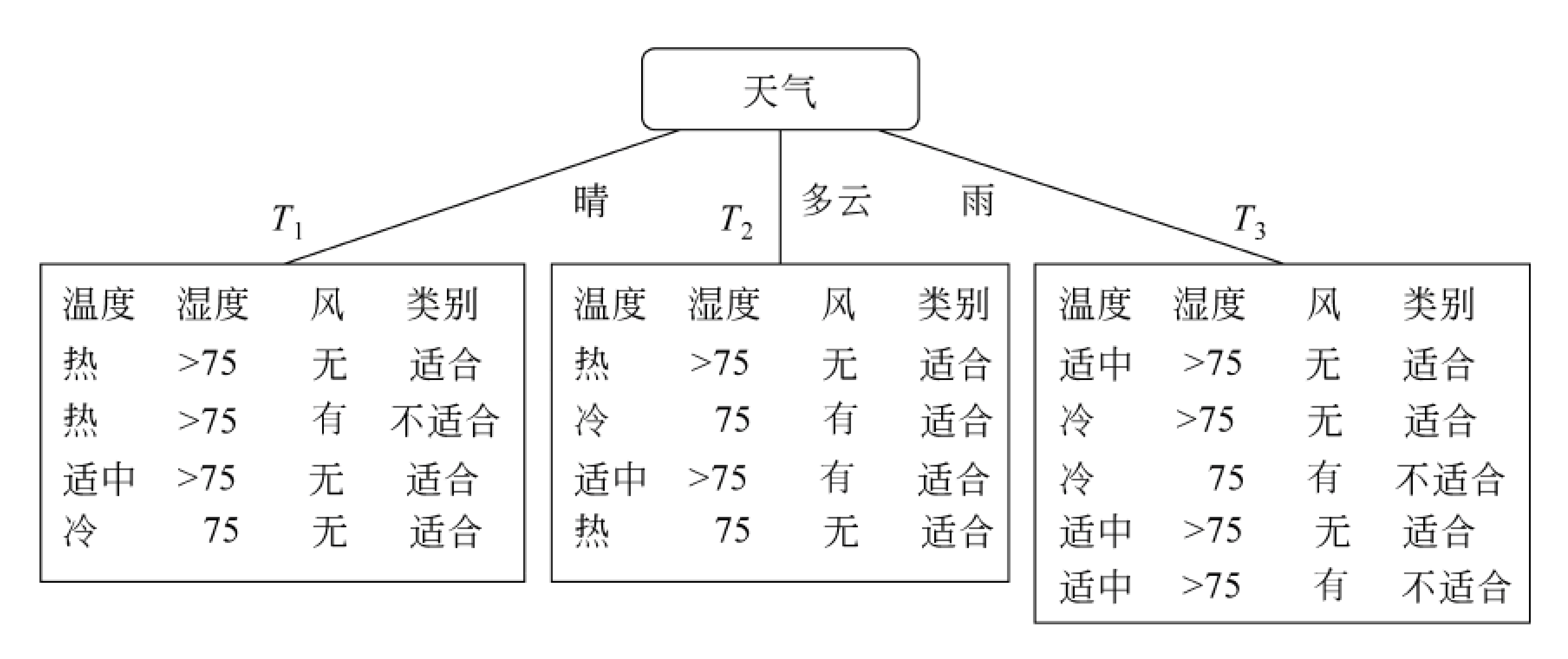


图 6-3　天气结点及其分支

下面说明各个子树的生成过程:

对于第一个子树 T_1,GainRatio(湿度)=1,GainRatio(温度)=0.0244,GainRatio(风)=0.0206。选择"湿度"作为分类属性,根据"湿度"的两个取值生成两个分支,得到两个叶结点。

第二个子树 T_2 中的所有样本都属于同一类(类别=适合),所以直接得到叶子结点。

对于第三个子树 T_3,GainRatio(湿度)=0.446,GainRatio(温度)=0.0206,GainRatio(风力)=1。选择"风力"成分支,得到两个叶结点。图 6-4 给出了最终生成的决策树。

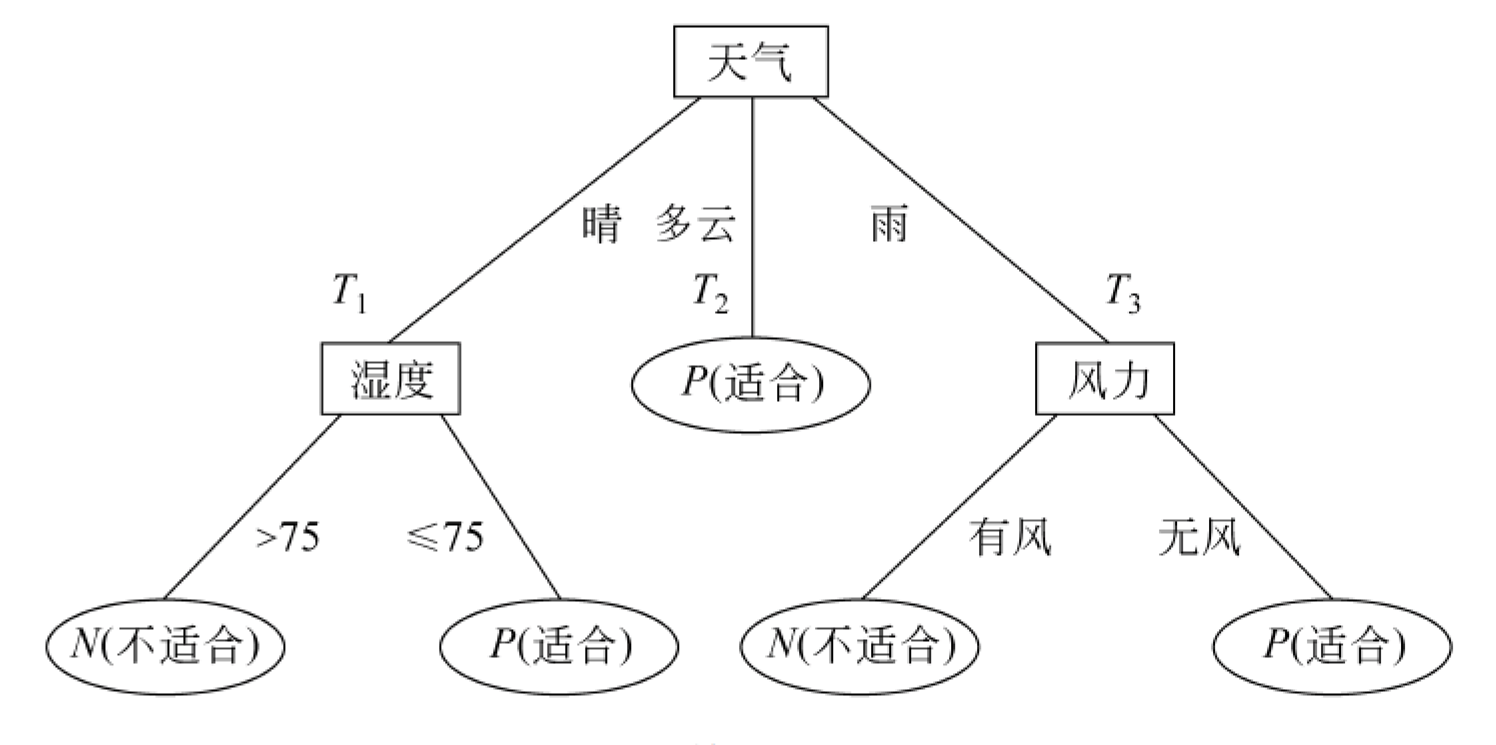


图 6-4　C4.5 算法形成的决策树

6.3 决策树剪枝

在创建决策树时，由于训练样本太少或数据中存在噪声和孤立点，许多分支反映的是训练数据中的异常现象，建立的决策树会过度拟合训练样本集。过度拟合是指推出过多与训练数据集相一致的假设，反而不具有很好的预测性能。剪枝是一种克服噪声的技术，同时它也能使树得到简化而变得更容易理解。通常，这种方法使用统计度量，剪去不可靠的分支，这将导致较快的分类，提高树独立于测试数据正确分类的能力。

有两种剪枝策略，先剪枝和后剪枝。先剪枝也有人称其为预剪枝，预剪枝技术限制决策树的过度生长，后剪枝技术则是待决策树生成后再进行剪枝。

6.3.1 先剪枝

最直接的先剪枝方法是事先限定决策树的最大生长高度，使决策树不能过度生长。这种停止标准一般能够取得比较好的效果。但指定树高度的方法要求用户对数据的取值分布有比较清晰的掌握，并需要对参数值进行反复尝试，才能给出一个比较合理的树高度阈值。

常用的方法是采用统计意义下的 χ^2 检验、信息增益等度量，评估每次结点分裂对系统性能的增量，如果结点分裂的增量小于预先给定的阀值，则不对该结点进行扩展。如果在最好情况下的扩展增益都小于阈值，即使有些结点的样本不属于同一类，算法也可以终止。困难的是选取合适的阀值，阀值高可能导致决策树过于简单，阀值低会对决策树化简不充分。

先剪枝算法有可能过早停止树的生长而存在视野效果问题，但该算法效率高，适合于规模大的问题。

6.3.2 后剪枝

后剪枝技术允许决策树过度生长，然后根据一定的规则，剪去那些不具有一般代表性的结点和分支。可以采用自上而下的顺序或自下而上的顺序进行剪枝。

代价复杂性剪枝算法是后剪枝方法之一，其思路是：最下面的未被剪枝的结点称为树叶，并用它先前分支中最频繁的类标记。对于树中每个非树叶结点，计算该结点上的子树被剪枝可能出现的期望错误率；然后使用每个分支的错误率，结合每个分支观察的权重评估，计算不对该结点剪枝的期望错误率。如果剪去该结点导致较高的期望错误率，则保留该子树；否则剪去该子树。产生一组逐渐被剪枝的树之后，使用一个独立的测试集评估每棵树的准确率，就能得到具有最小期望错误率的决策树。

剪枝之后决策树的叶结点不再只包含一类实例，结点有一个类分布描述，即该叶结点属于某类的概率。

可以将先剪枝和后剪枝算法交叉使用，后剪枝所需的计算比先剪枝多，但能产生更可靠的树。

6.4 由决策树提取分类规则

从决策树提取规则可以分为获得简单规则和获得精简规则属性两个步骤。

6.4.1 获得简单规则

对于生成好的决策树，可以直接从中获得规则。从根到叶的每一条路径都可以是一条规则。规则采用 IF…THEN 的形式表示。例如，如图 6-5 所示的决策树。

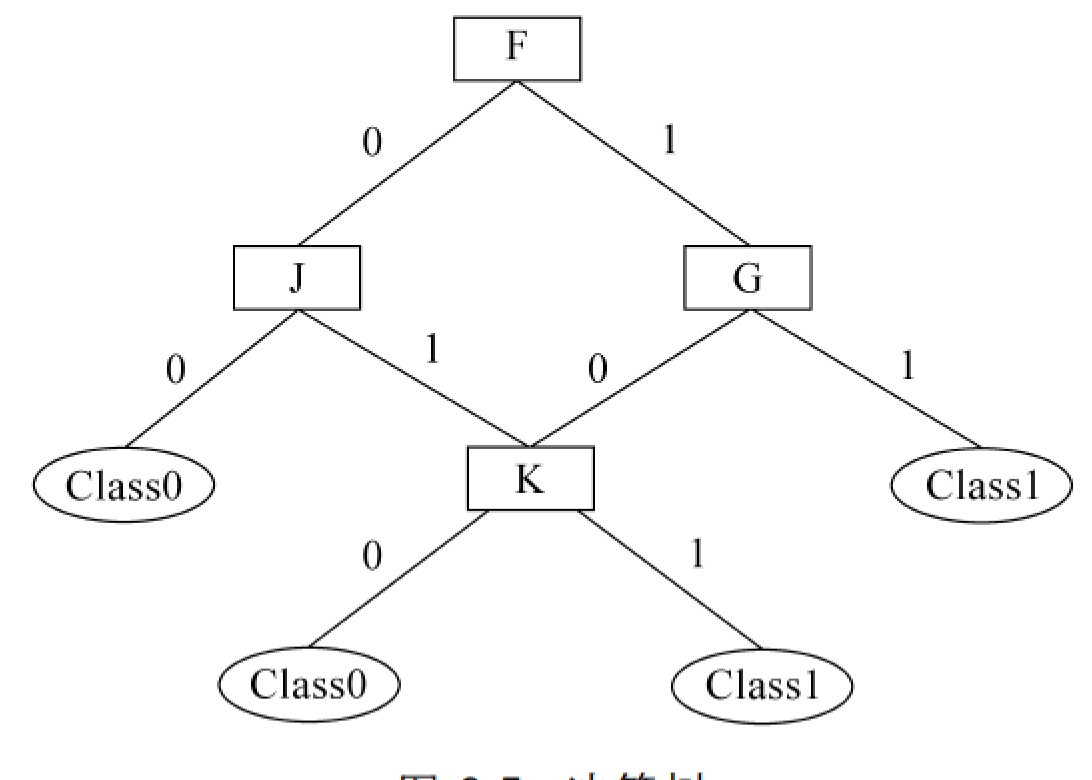


图 6-5 决策树

通过图 6-5，可以得到如下规则：

```
IF F=1,G=0,K=1 THEN Class1
```

6.4.2 精简规则属性

在不影响规则预测效果的情况下，可以删除一些分支，减少过度拟合，同时获得更容易理解的精简规则。

设规则的形式为 R：

```
IF A THEN Class C
```

精简之后的规则为 R^-：

```
IF A- THEN Class C
```

其中 A^- 是从 A 中删除某些条件 X 之后的形式。这样，规则 R^- 覆盖的实例可以分为以下 4 个部分，如表 6-3 所示。

表 6-3 规则 R^- 覆盖实例表

	Class C	其他类
满足条件 A	$Y1$	$E1$
满足条件 A^-，但不满足 X	$Y2$	$E2$

规则 R 覆盖了 $Y1+E1$ 个实例，其中误判数目为 $E1$，规则 R^- 覆盖了 $Y1+E1+Y2+E2$。规则 R 的误判概率为 $P(E1)=E1/(E1+Y1)$，规则 R^- 的误判概率为 $P(E1+E2)=(E1+E2)/(Y1+Y2+E1+E2)$。如果 $P(E1)\geqslant P(E1+E2)$，则可以从条件 A 中删除条件项 X。

如何获得最优条件集是全局优化问题。对于有很多决策属性的情况，这样做会非常耗

时。为此，Quinlan 提出了一种贪婪搜索方法，即每次从条件集合中删除一个对预测效果影响最小的条件，如果删除该条件后，误判概率减小了，则继续上述过程。如果误判概率增加了，则不能够删除该条件，而整个精简过程也同时结束。

6.5 利用 SQL Server 2005 进行决策树挖掘

Microsoft 决策树(Microsoft decision trees)是由 Microsoft 研究院开发的混合型的决策树算法，主要用来完成分类、回归和关联工作。其英文名称中出现的是 Trees，而不是 Tree，原因如下：

首先，在决策树算法中，可以通过设置不同的参数得到不同形状的决策树。这些树的生成实际是基于不同的决策树算法的。

其次，一个决策树模型可能包含多个树，有时甚至会有成百上千个。

Microsoft 决策树算法是由 Microsoft SQL Server 2005 Analysis Service(SSAS)提供的分类和回归算法，用于对离散和连续属性进行预测性建模。下面通过一个实例来说明如何利用 SQL Server Business Intelligence Development Studio 进行决策树数据挖掘。

6.5.1 数据准备

在进行数据挖掘前，需要准备好数据挖掘使用的数据库。本章实例沿用书中第 5 章的“商业银行信贷”数据库，主要数据库表结构和数据挖掘使用的主要数据视图结构，参见第 5 章图 5-5～图 5-7，在此不再重述。

为了得到更简洁的决策树结构，首先对该数据库中“t_dm”表的次级、可疑、损失、余额和正常 5 个字段进行了更新处理，SQL 处理语句如下：

```
Update 贷款余额表 set 次级=1 where 次级>0.0
Update 贷款余额表 set 可疑=1 where 可疑>0.0
Update 贷款余额表 set 损失=1 where 损失>0.0
Update 贷款余额表 set 余额=1 where 余额>0.0
Update 贷款余额表 set 正常=1 where 正常>0.0
```

6.5.2 挖掘模型设置

数据准备好之后，需要定义和设置挖掘模型，包括两个方面：指定列的用法和设置挖掘参数。其中挖掘参数设置本实例使用默认设置。列的用法如表 6-4 所示。

表 6-4 列的用法

字段名称	列的用法	字段名称	列的用法
ID	键列	客户状态	输入列
客户名称	输入列	重点标志	输入列
客户类型	输入列	可疑	预测列
经济性质	输入列	次级	预测列
隶属关系	输入列	正常	预测列
关注	输入列	余额	预测列
法人资格	输入列	损失	预测列

6.5.3 挖掘流程

以上准备工作完成之后就可以进行决策树数据挖掘。整个流程分为数据源设置、数据源视图设置和创建数据挖掘模型三个部分。在 SQL Server Business Intelligence Development Studio 工具中都有专门的设置向导帮助完成，在此处仅介绍数据挖掘模型的创建这一流程。

数据挖掘模型的创建由以下步骤完成。

(1) 右击项目 Decision Tree 下的“挖掘结构”，选择“新建挖掘结构”，打开“数据挖掘向导”对话框，单击“下一步”按钮，切换到“选择定义方法”页面，单击“下一步”按钮，进行下一步操作。

(2) 在如图 6-6 所示的对话框中，下拉列表框中选取“Microsoft 决策树”选项，单击“下一步”按钮，进行下一步操作。

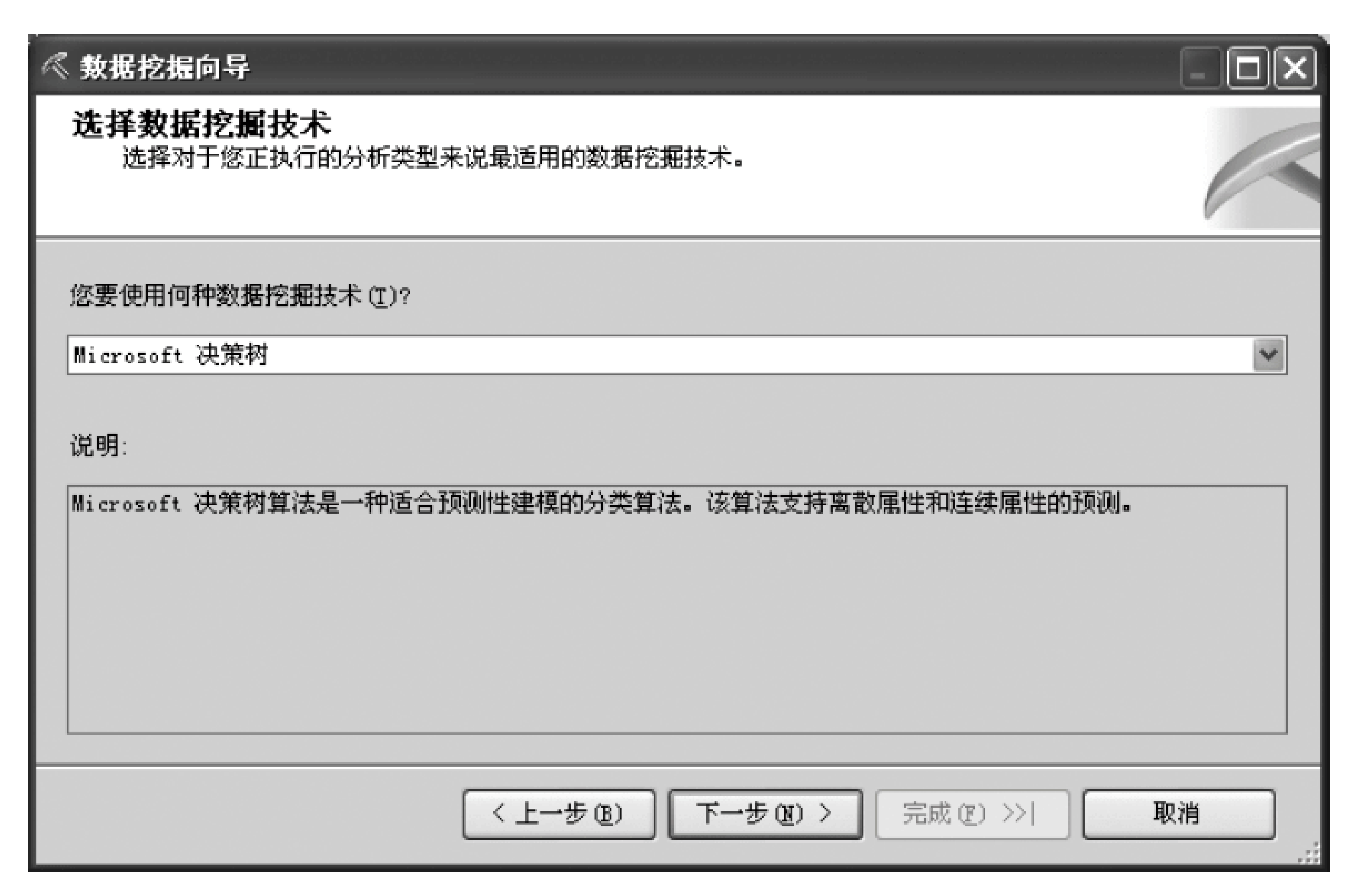


图 6-6 选择数据挖掘技术

(3) 如图 6-7 所示，在“选择数据源视图”页面的“可用数据源视图”列表中显示了前面步骤创建的 Bank 数据源视图，选中该视图选项，单击“下一步”按钮，进行下一步操作。

(4) 如图 6-8 所示，在“指定表类型”页面中可以看到 Bank 数据源视图包含的数据表，勾选“t_dm”选项右边的“事例”复选框，可以将其定义为事例表；单击“下一步”按钮，进行下一步操作。

(5) 如图 6-9 所示，在“指定定型数据”页面显示了挖掘模型结构，在各个选项右边勾选不同的复选框，可以将不同的表和列设置为键表、键列、输入列和可预测列等，各种本步骤的设置参考表 6-4 来完成，然后单击“下一步”按钮，进行下一步操作。

(6) 如图 6-10 所示，经过“检测”将指定数字列，即“次级”、“关注”、“可疑”、“损失”、“余额”和“正常”的连续值转换成离散值，即 0 或 1，与第 6.5.1 小节中做的数据处理对应起来。在“指定列的内容和数据类型”页面中显示了指定“ID”的内容类型为 Key，“余额”的内容类型为 Continuous，其余列内容类型均为 Discrete；ID 的数据类型为 long，“法人资格”、“经济性质”、“客户类型”、“客户状态”、“隶属关系”和“重点标志”的数据类型为 Text，其余各列数据类型均为 Double，单击“下一步”按钮，进行下一步操作。



图 6-7　选择数据源视图



图 6-8　指定表类型

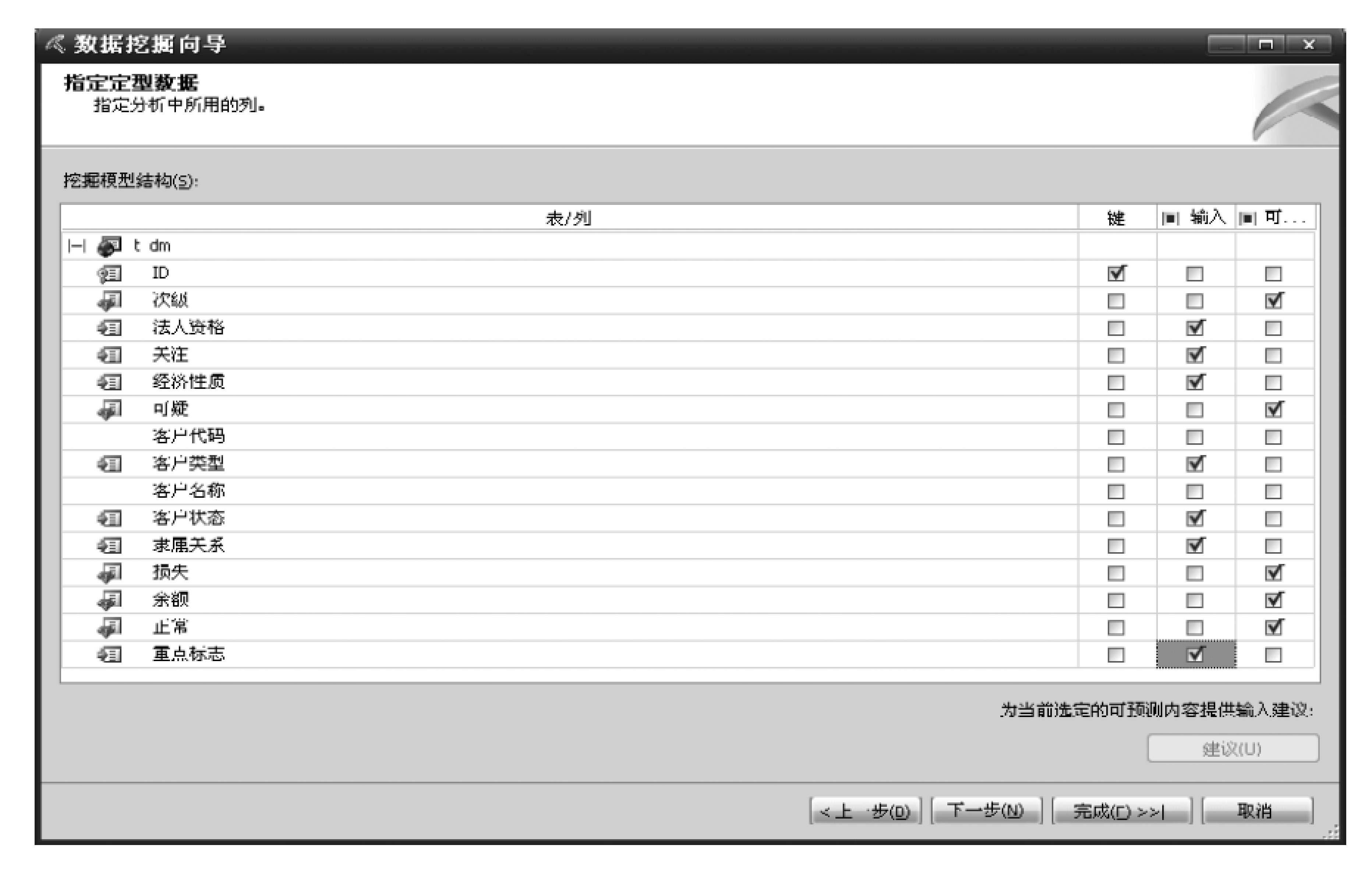


图 6-9　指定定型数据



图 6-10　指定列的内容和数据类型

(7) 如图 6-11 所示，在“完成向导”页面中将数据挖掘结构命名为 tDm1，单击“完成”按钮，完成挖掘结构的创建。



图 6-11　完成数据挖掘结构的创建

6.5.4　挖掘结果分析

(1) 单击“挖掘模型”选项卡下的“决策树”，其结果如图 6-12 所示，图中决策树是对预测列“次级”产生的，其中，深色代表 1-类，浅色代表 0-类。事实上，选择其余预测列“可疑”、“正常”、“余额”、“损失”均会产生与该预测列相应的决策树，在此，不再展开。

(2) 单击“挖掘模型”选项卡下的“依赖关系网络”，其结果如图 6-13 所示。图中有向边的终点是预测列，各有向边的起点是对该预测列有影响的输入列。

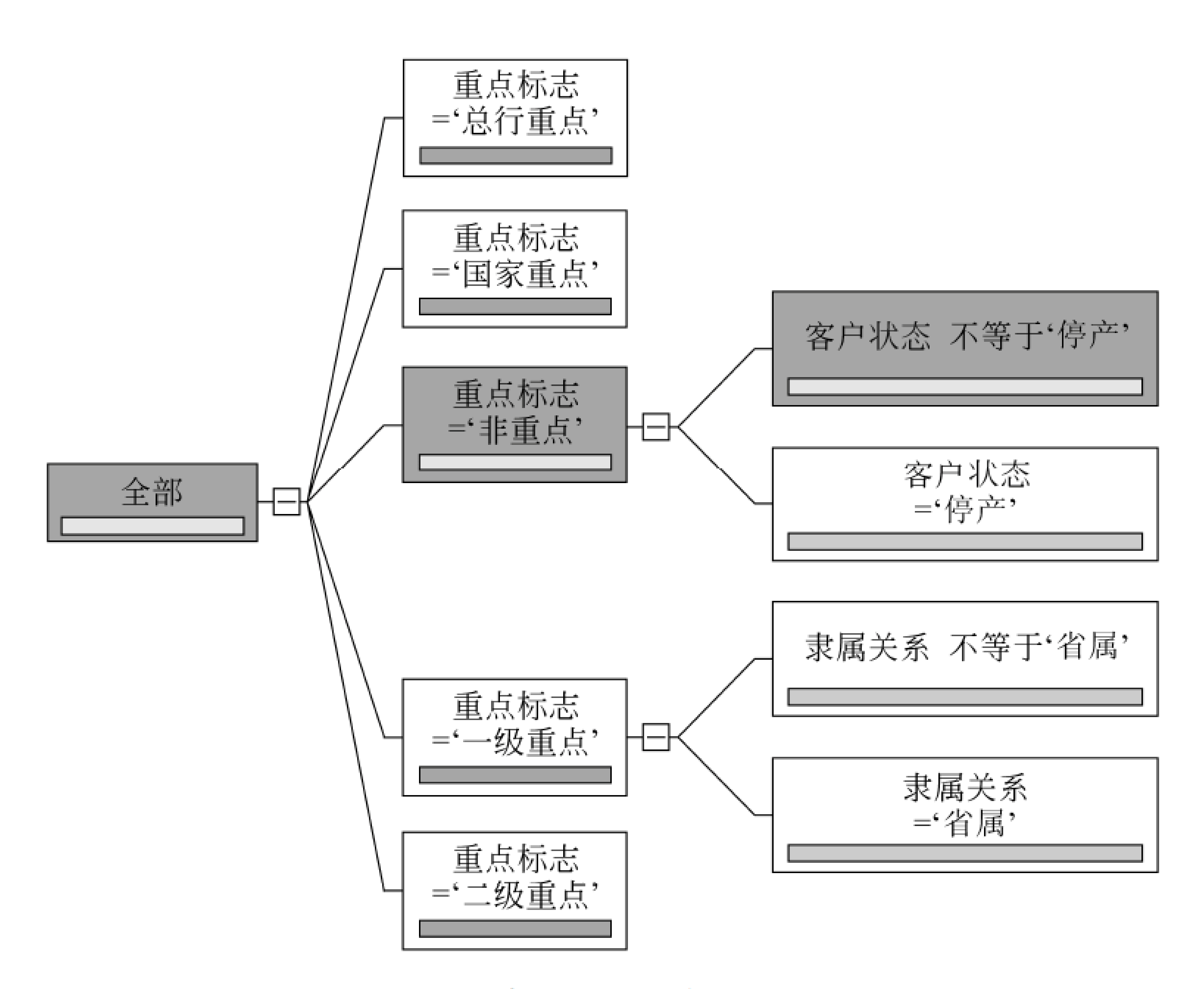


图 6-12　挖掘得到的“次级”决策树

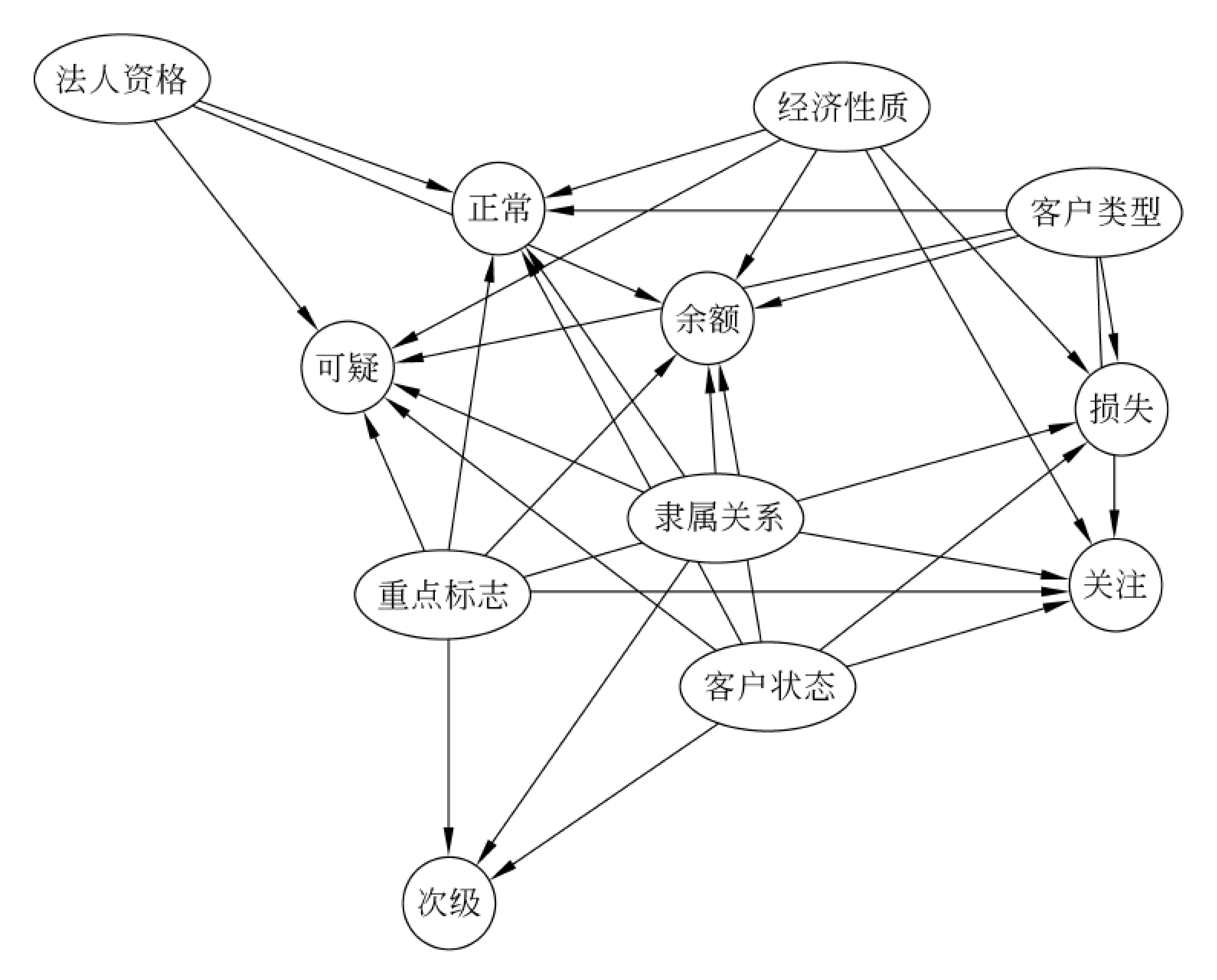


图 6-13　挖掘得到的依赖关系图

通过选择依赖关系图中的某个结点，可以突显出其依赖关系。例如，选中结点“余额”，可疑显示出其影响因素有“客户类型”、“经济性质”、“正常”、“隶属关系”、“重点标志”和“法人资格”，如图 6-14 所示。

通过选择链接程度，还可以找出对图中结点受影响的程度。以“余额”结点为例，可以得到受影响最大的结点是“重要标志”，如图 6-15 所示。该结点可以得到如表 6-5 所示的依赖关系表。

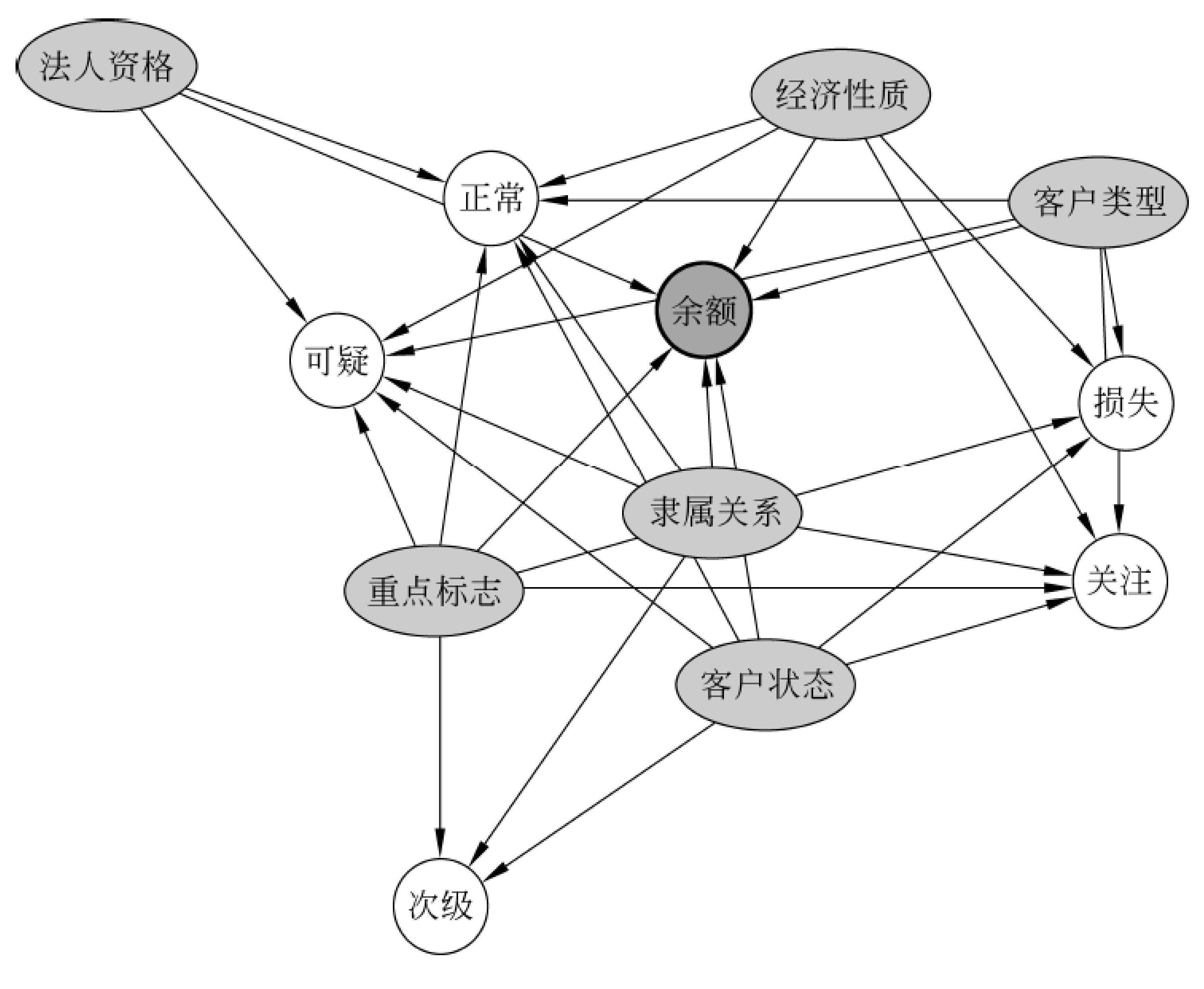


图 6-14 “余额”结点的依赖关系图

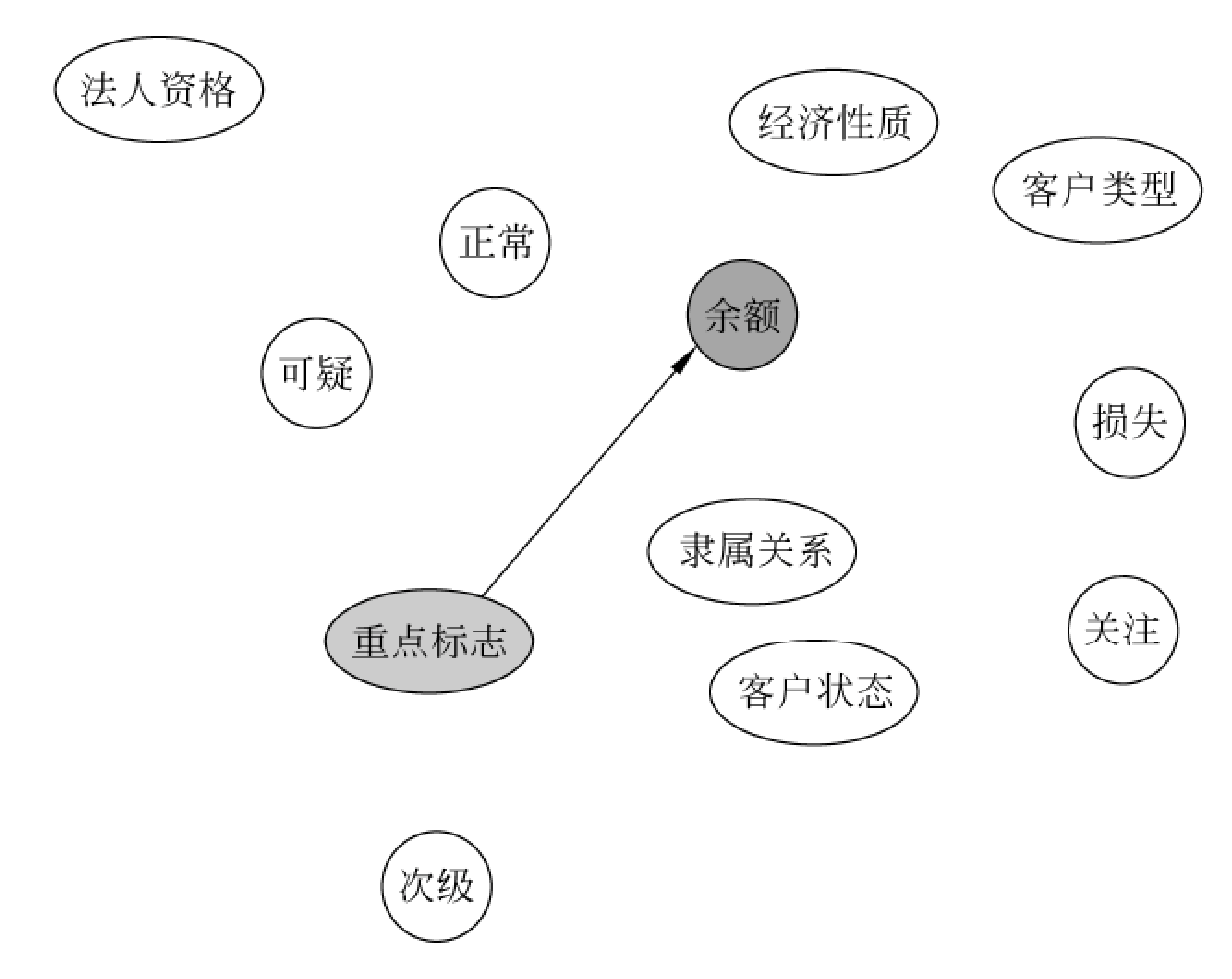


图 6-15 与“余额”结点链接强度最强结点示意图

表 6-5 “余额”结点依赖关系表

预测结点	决策结点	依赖关系强弱顺序
余额	重要标志	1
	隶属关系	2
	客户类型	3
	经济性质	4
	客户状态	5
	法人资格	6

6.5.5 挖掘性能分析

（1）单击“挖掘准确性图表”中“列映射”选项卡，选择已建立的挖掘结构 tDm1，选择 t_dm 作为输入表，筛选“余额”字段作为输入数据，选择“次级”作为提升图输出，如图 6-16 所示。

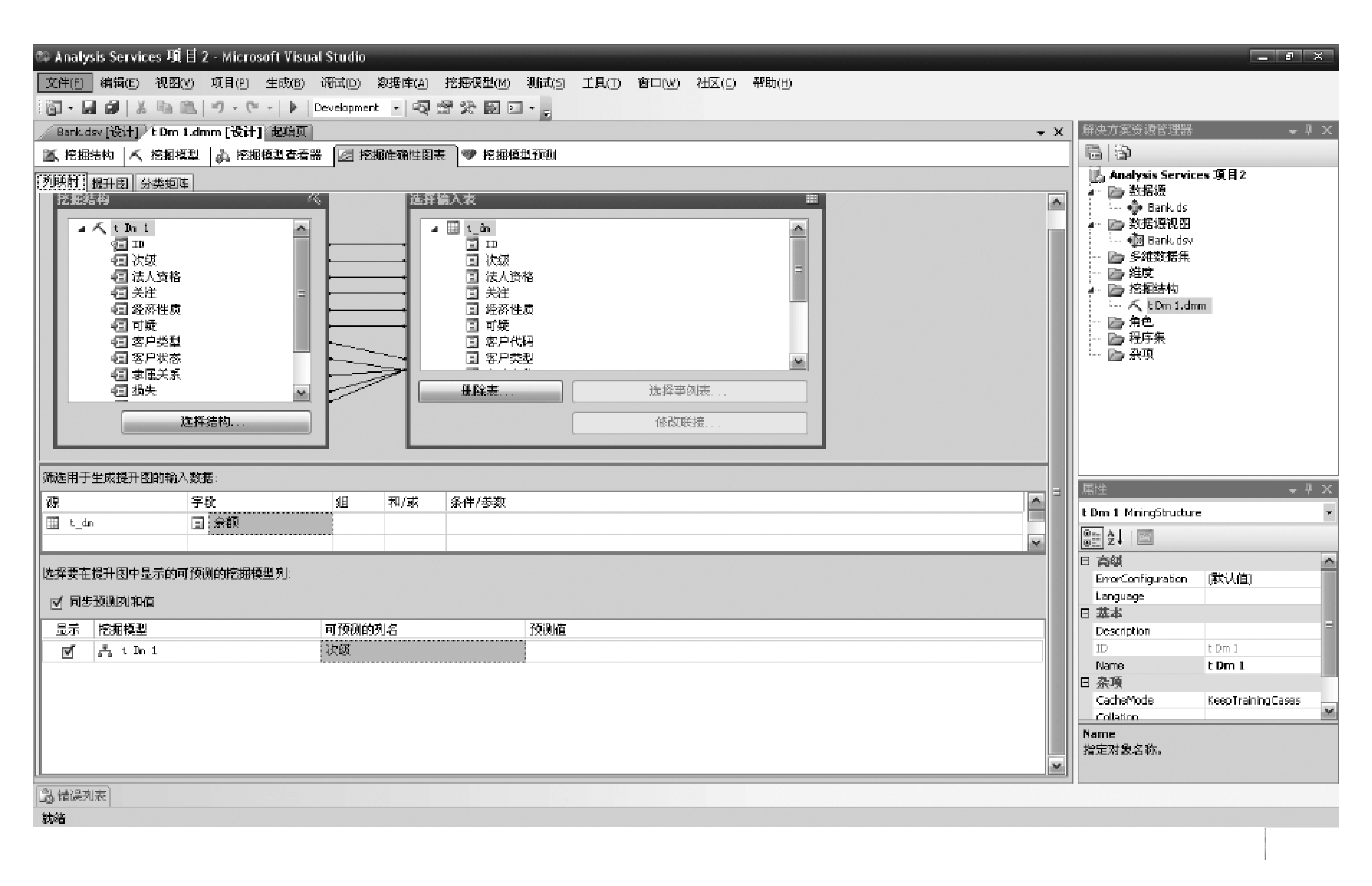


图 6-16 列映射图

（2）单击“挖掘准确性图表”中“提升图”选项卡，显示如图 6-17 所示的提升图，这是以属性“次级”为输出属性得到的，该图表明预测模型与理想模型是比较匹配的。

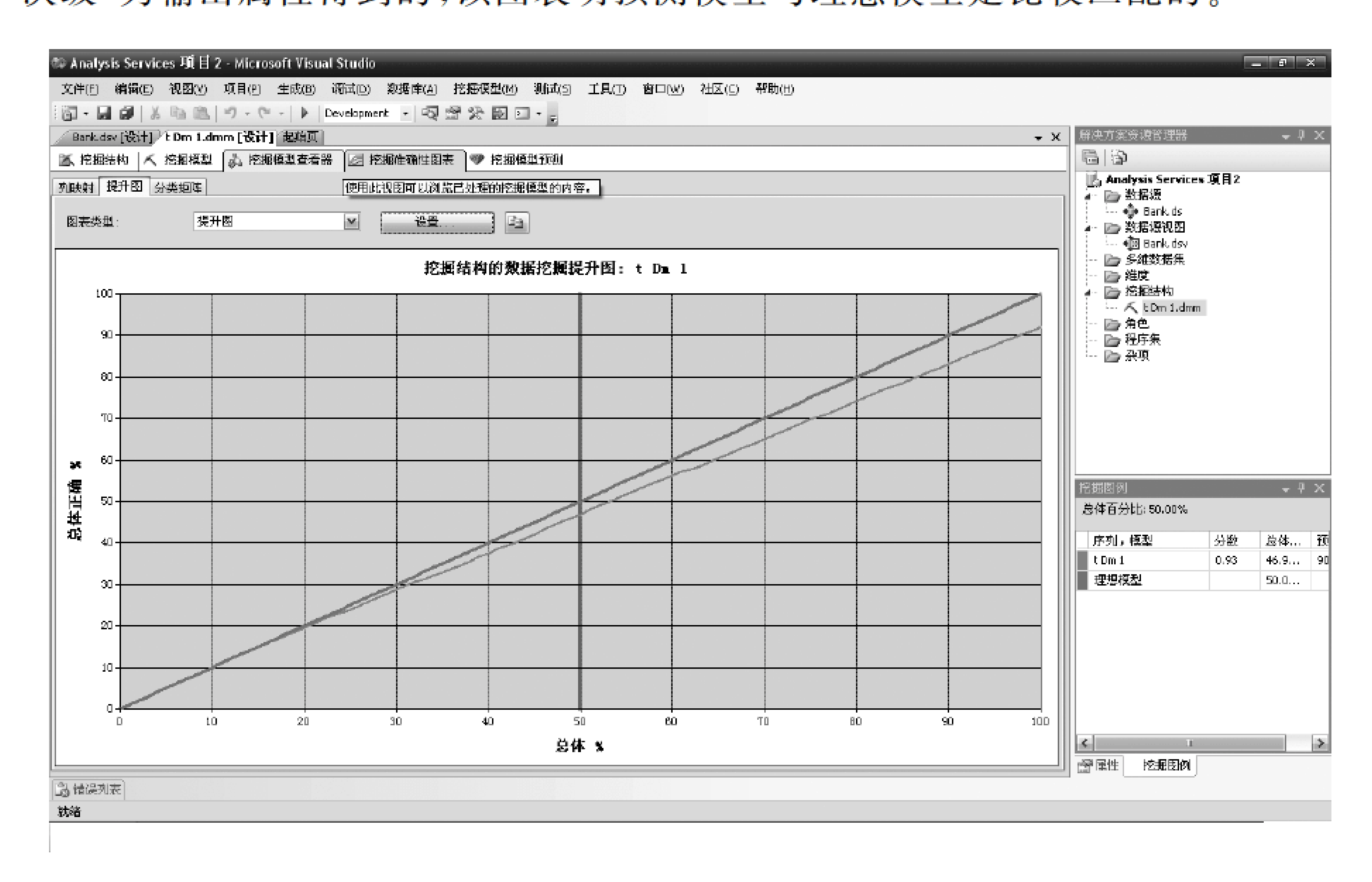


图 6-17 属性“次级”的预测提升图

小结

本章主要介绍了决策树算法的各种相关知识，首先对决策树算法的信息论原理进行了介绍，并介绍和分析了常用的决策树算法；其次对提高决策树算法生成的决策树的分类效果的决策树剪枝技术进行了讲解，最后，采用 SQL Server 2005 数据挖掘工具实现了一个银行客户分析的实例，进一步加深对决策树算法的认识和理解。

习题 6

1. 概率分布[0：0625；0：0625；0：125；0：5]的熵是多少？

2. 汽车保险例子。假定训练数据库具有两个属性：年龄和汽车的类型。

年龄——序数分类。

汽车类型——分类属性。

类——L：低(风险)，H：高(风险)。

年龄	汽车类型	类
>21	Maruti	L
>21	Hyundai	H
<21	Maruti	H
<21	Indica	H
>21	Maruti	L
>21	Hyundai	H

使用 ID3 算法做出它的决策树。

3. 简述 ID3 和 C4.5 算法之间的异同。

4. 简述决策树剪枝的步骤。

5. 练习 SQL Server 2005 决策树挖掘模型的构建。

第 7 章　统计学习方法

统计学是搜集、展示、分析及解释数据的科学，数据挖掘大部分核心功能的实现都以计量和统计分析方法作为支撑。许多成熟的统计方法构成了数据挖掘的核心内容。

常见的统计分析方法有回归分析(多元回归、自回归)、判别分析(贝叶斯判别、费歇尔判别、非参数判别)以及探索性分析(主元分析、相关分析)等。下面主要介绍朴素贝叶斯分类、贝叶斯信念网络、EM 算法和回归分析方法。

7.1　朴素贝叶斯分类

朴素贝叶斯分类器是贝叶斯分类模型中一种最简单、有效而且在实际使用中很成功的分类器，朴素贝叶斯分类基于贝叶斯定理，在实际运用中降低了贝叶斯网络构建的复杂性。分类算法的比较研究发现，朴素贝叶斯分类算法可以与判定树和神经网络分类算法相媲美，用于大型数据库分析；朴素贝叶斯分类也已表现出高准确率与高速度，而且已经成功地应用于聚类、分类等数据挖掘任务中。

7.1.1　贝叶斯定理

设 X 是类标号未知的数据样本。设 H 为某种假定，如数据样本 X 属于某特定的类 C。对于分类问题，希望确定 $P(H/X)$——给定观测数据样本 X，假定 H 成立的概率。

$P(H/X)$称为条件 X 下，H 的后验概率。例如，假定数据样本空间由水果组成，用它们的颜色和形状描述；若 X 表示一个样本是红色和圆的，H 表示假定这个样本是苹果，则 $P(H/X)$反映当看到一个样本是红色并是圆的时，对这个样本是苹果的确信程度。

$P(H)$称为 H 的先验概率。对于上面的例子，它是任意给定的数据样本为苹果的概率，而不管数据样本看上去如何。

后验概率比先验概率基于更多的信息(如，背景知识)。

假设先验概率 $P(X)$、$P(H)$和后验概率 $P(X/H)$可以由给定的数据计算，贝叶斯定理提供了一种由 $P(X)$，$P(H)$和 $P(X/H)$计算后验概率 $P(H/X)$的方法，即贝叶斯公式：

$$P(H/X)=\frac{P(X/H)P(H)}{P(X)} \tag{7-1}$$

例如，假设 A 为事件“产品合格”，B 为“机器工作正常”，现给出以下概率：

机器工作正常，生产产品合格的概率为 $P(A/B)=0.95$；

机器正常工作，生产产品不合格的概率为 $P(\overline{A}/B)=1-P(A/B)=0.05$；

机器不正常工作时，生产产品合格的概率为 $P(A/\overline{B})=0.1$；

机器正常工作的概率，即 $P(B)=0.9$。

已知生产了一个不合格产品，通过计算 $P(B/\overline{A})$的概率，并与 $P(\overline{B}/\overline{A})$进行比较，可以从概率角度判断机器正常工作还是不正常工作的可能性大。

利用贝叶斯定理进行计算，过程如下：

$$P(B/\bar{A})=\frac{P(\bar{A}/B)\times P(B)}{P(\bar{A})}=\frac{P(\bar{A}/B)\times P(B)}{P(\bar{A}/B)\times P(B)+P(\bar{A}/\bar{B})\times P(\bar{B})}$$

$$=\frac{0.05\times 0.9}{0.05\times 0.9+0.9\times 0.1}=0.333$$

$$P(\bar{B}/\bar{A})=1-0.333=0.667$$

可以看出 $P(B/\bar{A})<P(\bar{B}/\bar{A})$，由此得出结论为，生产出一个不合格产品时，机器不正常工作的概率比较大。

7.1.2 朴素贝叶斯分类

朴素贝叶斯分类过程如下。

(1) 每个数据样本用一个 n 维特征向量 $X=(x_1,x_2,\cdots,x_n)$ 表示，属性 $A_1,A_2,\cdots,A_n$ 描述对样本的 n 个度量。

(2) 假定有 m 个类 $C_1,C_2,\cdots,C_m$。给定一个未知的数据样本 X（即没有类标号），分类法将预测 X 属于具有最高后验概率（条件 X 下）的类。即朴素贝叶斯分类将未知的样本分配给类 C_i，当且仅当：

$$P(C_i/X)>P(C_j/X),\quad 0\leqslant j\leqslant m,j\neq i \tag{7-2}$$

时，于是可以最大化 $P(C_i/X)$，其中

$$P(C_i/X)=\frac{P(X/C_i)P(C_i)}{P(X)}$$

(3) 由于 $P(X)$ 对于所有类为常数，只需要 $P(X/C_i)P(C_i)$ 最大即可。

若类的先验概率未知，则通常假定这些类是等概率的，即 $P(C_1)=P(C_2)=\cdots=P(C_m)$。据此只需对 $P(X/C_i)$ 最大化。

若类的先验概率已知，则最大化 $P(X/C_i)p(C_i)$。类的先验概率可以用 $P(C_i)=s_i/s$ 计算，其中 s_i 是类 C_i 中的训练样本数，而 s 是训练样本总数。

(4) 给定具有许多属性的数据集，计算 $P(X/C_i)$ 的开销可能非常大。为降低计算 $P(X/C_i)$ 的开销，可以做类条件独立的朴素假定，即给定样本的类标号，假定属性值条件地相互独立，即在属性间不存在依赖关系。这样，

$$P(X/C_i)=\prod_{k=1}^{n}P(x_k/C_i) \tag{7-3}$$

概率 $P(x_1/C_i),P(x_2/C_i),\cdots,P(x_n/C_i)$ 可以由训练样本估值。

① 如果 A_k 是离散型属性，则 $P(x_k/C_i)=s_{ik}/s_i$；其中 s_{ik} 是在属性 A_k 上具有值 x_k 的类 C_i 的训练样本数，而 s_i 是 C_i 中的训练样本数。

② 如果 A_k 是连续型属性，则通常假定该属性服从高斯分布。因而，

$$P(x_k/C_i)=g(x_k,\mu_{C_i},\sigma_{C_i})=\frac{1}{\sqrt{2\pi}\,\sigma_{C_i}}\mathrm{e}^{-\frac{(x-\mu_{C_i})^2}{2\sigma_{C_i}^2}} \tag{7-4}$$

其中，给定类 C_i 的训练样本属性 A_k 的值，$g(x_k,\mu_{C_i},\sigma_{C_i})$ 是属性 A_k 的高斯密度函数，而 μ_{C_i}，σ_{C_i} 分别为平均值和标准差。

(5) 对每个类 C_i，计算 $P(X/C_i)P(C_i)$。样本 X 被指派到类 C_i，当且仅当：

$$P(X/C_i)P(C_i) > P(X/C_j)P(C_j), \quad 1 \leqslant j \leqslant m, j \neq i$$

换言之，X 被指派到使 $P(X/C_i)P(C_i)$最大的类 C_i。

例如，给定表 7-1 所示的训练数据，数据样本用属性 age，income，student 和 credit_rating 描述。类标号属性 buys_computer 具有两个不同值(即{ yes，no})。给定一个没有类标号的数据样本 X=(age="<=30"，income="medium"，student="yes"，credit _ rating ="fair")，下面使用朴素贝叶斯分类预测这个数据样本的类标号。

表 7-1　AllElectronics 顾客数据库训练数据元组

RID	age	income	student	Credit_rating	Class:buys_computer
1	<=30	high	no	fair	No
2	<=30	high	no	excellent	No
3	31…40	high	no	fair	Yes
4	>40	medium	no	fair	Yes
5	>40	low	yes	fair	Yes
6	>40	low	yes	excellent	No
7	31…40	low	yes	excellent	Yes
8	<=30	medium	no	fair	No
9	<=30	low	yes	fair	Yes
10	>40	medium	yes	fair	Yes
11	<=30	medium	yes	excellent	Yes
12	31…40	medium	no	excellent	Yes
13	31…40	high	yes	fair	Yes
14	>40	medium	no	excellent	No

设 C_1 对应于类 buys_computer="yes"，而 C_2 对应于类 buys_computer="no"。根据前面的讲述，需要最大化 $P(X/C_i)P(C_i)$，$i=1,2$。

每个类的先验概率 $P(C_i)$可以根据训练样本计算：

P(buys_computer="yes")=9/14=0.643

P(buys_computer="no")=5/14=0.357

为计算 $P(X/C_i)$，$i=1,2$，计算下面的条件概率：

P(age="<30"|buys_computer="yes")=2/9=0.222

P(age="<30"|buys_computer="no")=3/5=0.600

P(income="medium"|buys_computer="yes")=4/9=0.444

P(income="medium"|buys_computer="no")=2/5=0.400

P(student="yes"|buys_computer="yes")=6/9=0.667

P(student="yes"| buys_computer="no")=1/5=0.200

P(credit_rating="fair"|buys_computer="yes")=6/9=0.667

P(credit_rating="fair"|buys_computer="no")=2/5=0.400

使用以上概率，可以得到：

P(X|buys_computer="yes")=0.222×0.444×0.667×0.667=0.044

P(X|buys_computer="no")=0.600×0.400×0.200×0.400=0.019

P(X|buys_computer="yes") P(buys_computer="yes")=0.044×0.643=0.028

P(X|buys_computer="no") P(buys_computer="no")=0.019×0.357=0.007

因此，对于样本 X，朴素贝叶斯分类预测 buys_computer="yes"。

7.2 贝叶斯信念网络

7.2.1 贝叶斯信念网络

贝叶斯信念网络也称为信念网络、贝叶斯网络或概率网络，是一种不确定知识表达和推理领域最有效的理论模型。这种理论于 1985 年由 Judea Pearl 首先提出，它的理论基础是基于后验概率的贝叶斯定理。

朴素贝叶斯假定类条件独立，即给定元组的类标号，假定属性的值可以有条件地相互独立。然而，在实践中，变量之间的依赖是可能存在的，属性集之间也并不是完全独立的，在这种情况下，朴素贝叶斯不能很好地对样本进行分类，贝叶斯网络的提出很好地解决了数据集之间的相关情况，它使用网络结构将不确定的事件形式化地表示出来，并可以用于分类、聚类、预测等关系的分析。

贝叶斯信念网络是基于概率推理的图形化网络，由一个有向无环图（directed acyclic graph，DAG）表示，图中的每个结点代表一个随机变量，结点间的有向边代表了结点间的概率依赖。

如果一条有向边由结点 A 指向结点 B，则称结点 A 是结点 B 的双亲，结点 B 是结点 A 的后代。

对于每个变量，信念网络有一个条件概率表（conditional probability table，CPT）。变量 Y 的 CPT 说明条件分布 $P(Y|\text{Parents}(Y))$，其中 Parents(Y)是 Y 的双亲。

贝叶斯网络的一个重要性质是：给定其双亲，每个变量有条件地独立于图中它的非后代。

一个简单的贝叶斯网络由图 7-1 给出，它对下雨(R)引起草地变湿(W)进行建模。天下雨的可能性为 40%，并且下雨时草地变湿的可能性为 90%，也许 10%的时间雨下得不长，不足以让草地被淋湿；存在 20%的可能性草地变湿而实际上并没有下雨，例如使用喷水器时。

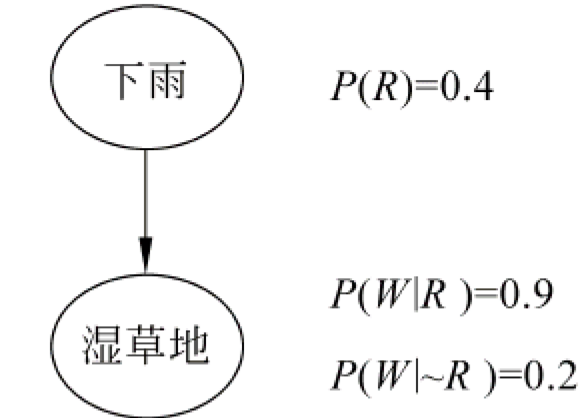


图 7-1 下雨使草地变湿的贝叶斯信念网络

在这个例子中，随机变量是二元的。可以看到以上 3 个概率就可以完全指定(R,W)的联合分布。因为 $P(R)=0.4$，则 $P(\sim R)=0.6$，类似地，$P(\sim W|R)=0.1$，$P(\sim W|\sim R)=0.8$。

7.2.2 贝叶斯网络的特点

(1) 贝叶斯网络通过网络结构图的方法来描述数据间的关系，语义清晰，可理解性强，有助于利用数据间的因果关系进行分析。

(2) 贝叶斯网络适合处理不完整的数据。对有属性遗漏的实例，可以通过对该属性的所有可能取值的概率求和或者积分来处理。

(3) 贝叶斯网络本身没有输入和输出的概念，各个结点的计算都是相对独立的，可以由上级向下级结点推理，也可以由下级向上级结点推理。

7.2.3 贝叶斯网络的应用

贝叶斯网络的主要应用包括诊断分析和预测推理，下面分别进行介绍。

1. 利用贝叶斯网络进行诊断分析

图 7-1 所示的贝叶斯网络实质上是一个因果图。解释草地变湿的主要原因是下雨。贝叶斯法则允许颠倒因果关系并做出诊断。例如，已知草地是湿的，则下过雨的概率可以计算如下：

$$P(R \mid W) = \frac{P(W \mid R)P(R)}{P(W)} = \frac{P(W \mid R)P(R)}{P(W \mid R)P(R) + P(W \mid \sim R)P(\sim R)}$$

$$= \frac{0.9 \times 0.4}{0.9 \times 0.4 + 0.2 \times 0.6} = 0.75$$

2. 利用贝叶斯网络进行预测推理

现在，假设把喷水器(S)作为草地变湿的另一个原因，如图 7-2 所示，此时结点 W 有两个双亲结点 R 和 S。现在可以计算喷水器开着草地会湿的概率，无须知道是否下过雨，这是一个预测推理。

$$\begin{aligned} P(W \mid S) &= P(W \mid R,S)P(R \mid S) \\ &\quad + P(W \mid \sim R,S)P(\sim R \mid S) \\ &= P(W \mid R,S)P(R) \\ &\quad + P(W \mid \sim R,S)P(\sim R) \\ &= 0.95 \times 0.4 + 0.9 \times 0.6 \\ &= 0.92 \end{aligned}$$

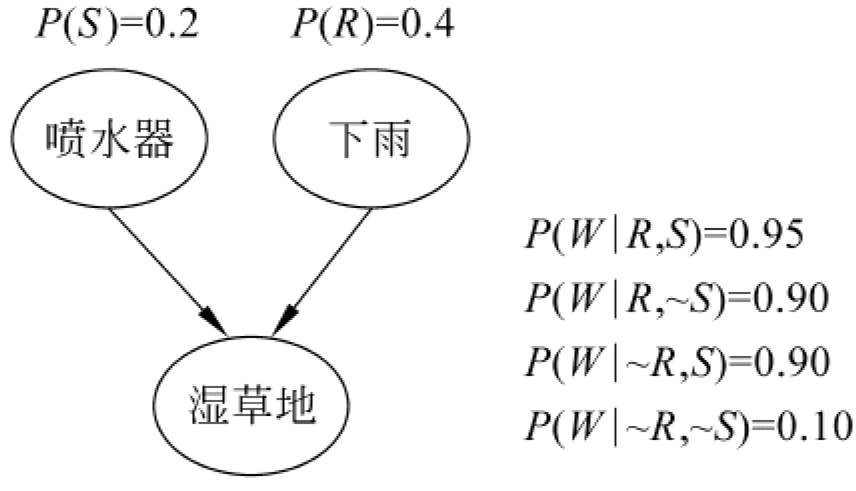


图 7-2 下雨和喷水器使草地变湿的贝叶斯信念网络

其中 $P(R|S)=P(R)$，这是因为根据图 7-2，R 和 S 是独立的。

7.3 EM 算法

如果知道总体 X 的分布类型，但分布中的参数未知，当需要确定未知参数时，可根据抽到的样本，对总体分布中的未知参数做出估计。极大似然估计就是一种常用的参数估计方法，它以观测值出现的概率最大作为准则。但是，如果训练数据集中的一些数据由于某些原因观测不完整，就必须借助于其他方法。

EM 算法是 Dempster，Laind，Rubin 于 1977 年提出的求参数极大似然估计的一种方法，它可以从非完整数据集中对参数进行极大似然估计，是一种非常简单实用的学习算法。这种方法可以广泛地应用于处理缺损数据、截尾数据，带有噪声的数据等各种不完整数据。

下面首先通过一个例子介绍 EM 算法的思想。

7.3.1 估计 k 个高斯分布的均值

考虑数据 D 是一个实例集合，它由 k 个不同正态分布的混合分布生成，如图 7-3 所示，其中 $k=2$ 而且实例为沿着 x 轴显示的点。每个实例通过两步骤过程形成。

(1) 随机选择 k 个正态分布中的一个。

（2）实例 x_i 按照此选择的分布生成。

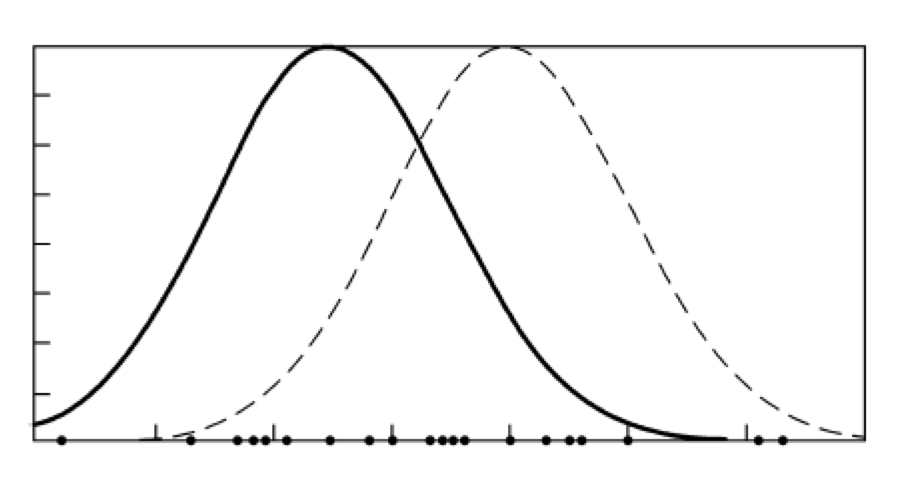


图 7-3　由两个具有相等方差的正态分布混合生成的实例

这一过程不断重复，生成一组数据点如图 7-3 所示。为使讨论简单化，考虑一个简单情形，即单个正态分布的选择基于均匀概率进行，并且 k 个正态分布有相同的方差 σ^2，且方差已知。输入一个假设 $h=\langle \mu_1 \cdots \mu_k \rangle$，它描述了 k 个分布中每一个分布的均值。现在的任务是，对这些均值找到一个极大似然假设，即一个使 $p(D/h)$ 最大化的假设 h。

当给定从一个正态分布中抽取的数据实例 $x_1, x_2, \cdots, x_m$ 时，很容易计算该分布均值的极大似然假设 μ_{ML}，可以得到：

$$\mu_{\mathrm{ML}} = \arg\min_{\mu} \sum_{i=1}^{m} (x_i - \mu)^2 \tag{7-5}$$

在此情况下，极大似然假设 μ_{ML} 等于样本均值，即

$$\mu_{\mathrm{ML}} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{7-6}$$

然而，问题涉及 k 个不同正态分布的混合，而且不知道哪个实例是哪个分布产生的，因此这是一个涉及隐藏变量的典型例子。可把每个实例的完整描述看做是三元组（x_i，z_{i1}，z_{i2}），其中 x_i 是第 i 个实例的观测值，z_{i1}，z_{i2} 表示两个正态分布中哪个被用于产生值 x_i。确切地讲 z_{ij} 在 x_i 由第 j 个正态分布产生时值为 1，否则为 0。这里，x_i 是实例描述中已观测到的变量，z_{i1}，z_{i2} 是隐藏变量。如果 z_{i1}，z_{i2} 的值可知，就可以求解均值 μ_1 和 μ_2。然而它们未知，因此选用 EM 算法。

EM 算法应用于 k 个正态分布的均值问题，目的是搜索一个极大似然假设，方法是根据当前假设（μ_1，μ_2）不断地再估计隐藏变量 z_{ij} 的期望值。然后用这些隐藏变量的期望值重新计算极大似然假设。

为了估计图中的两个均值，EM 算法首先将假设初始化为 $h=(\mu_1, \mu_2)$，其中 μ_1 和 μ_2 为任意的初始值。然后重复以下的两个步骤来估计 h，直到该过程收敛到一个稳定的值。

第 1 步，假定当前假设 $h=(\mu_1, \mu_2)$ 成立，计算每个隐藏变量 z_{ij} 的期望值 $E[z_{ij}]$。

第 2 步，假定每个隐藏变量 z_{ij} 所取的值为第 1 步中得到的期望值 $E[z_{ij}]$，计算一个新的极大似然假设，将假设 $h=(\mu_1, \mu_2)$ 替换为新的假设 $h'=(\mu_1', \mu_2')$，然后循环。

第 1 步要计算每个 z_{ij} 的期望值。此 $E[z_{ij}]$ 正是实例 x_i 由第 j 个正态分布产生的概率：

$$E[z_{ij}] = \frac{P(x = x_i \mid \mu = \mu_j)}{\sum_{n=1}^{2} P(x = x_i \mid \mu = \mu_n)} = \frac{\mathrm{e}^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}{\sum_{n=1}^{2} \mathrm{e}^{-\frac{1}{2\sigma^2}(x_i-\mu_n)^2}} \tag{7-7}$$

因此，第 1 步可由将当前值（μ_1，μ_2）和已知的 x_i 代入到上式中实现。

在第 1 步中，使用第 2 步中得到的 $E[z_{ij}]$ 来导出一新的极大似然假设 $h'=(\mu_1', \mu_2')$。这时的极大似然假设为

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}] x_i}{\sum_{i=1}^{m} E[z_{ij}]} \tag{7-8}$$

注意，此表达式类似于式 7-6 中的样本均值，它用于从单个正态分布中估计 μ。新的表达式只是对 μ_j 的加权样本均值，每个实例由第 j 个正态分布产生的期望值 $E[z_{ij}]$来权衡。

上面估计 k 个正态分布均值的例子描述了 EM 方法的要点，即当前的假设用于估计未知变量，而这些变量的期望值再被用于改进假设。可以证明，在此算法每一次循环中，EM 算法能使 $P(D/h)$增加，除非 $P(D/h)$已达到局部最大。因此该算法收敛于(μ_1,μ_2)的一个局部极大似然假设。

7.3.2 EM 算法的一般表述

上面的 EM 算法针对的是估计混合正态分布均值的问题。在上面的二均值问题中，感兴趣的参数 $\theta=(\mu_1,\mu_2)$，全部数据为三元组(x_i,z_{i1},z_{i2})，而只有 x_i 可观察到。

更为一般的情况是，EM 算法可用于许多问题框架，在这类问题中，需要估计一组描述基准概率分布的参数 θ，只给定了由此分布产生的全部数据中能观察到的一部分。

一般令 $X=(X_1,\cdots,X_m)$代表观察到的数据，并令 $Y=X\cup Z$ 代表全体数据。未观察到的 Z 可被看做一个随机变量，它的概率分布依赖于未知参数 θ 和已知数据 X。与此类似，Y 是一个随机变量，它是由随机变量 Z 来定义的。

下面将描述 EM 算法的一般形式，其中使用 h 来代表参数θ 的假设值，而 h'代表在 EM 算法的每次迭代中修改的假设。

EM 算法通过搜索使期望 $E[\ln P(Y/h')]$最大的 h'来寻找极大似然假设 h'。此期望值是在 Y 所遵循的概率分布上计算，此分布由未知参数 θ 确定。

首先，$P(Y/h')$是给定假设 h'下全部数据 Y 的似然度。其合理性在于我们要寻找一个 h'使函数值 $P(Y/h')$最大化。其次，使该量的对数 $\ln P(Y/h')$最大化也使 $P(Y/h')$最大化。第三，引入期望值 $E[\ln P(Y/h')]$是因为全部数据 Y 本身也是一个随机变量。已知全部数据 Y 是观察到的 X 和未观察到的 Z 的合并，必须在未观察到的 Z 的可能值上取平均并以相应的概率为权值。换言之，要在随机变量 Y 遵循的概率分布上取期望值 $E[\ln P(Y/h')]$。该分布由完全已知的 X 值加上 Z 服从的分布来确定。

Y 遵循的概率分布，一般来说是未知的，它由待估计的 θ 参数确定。EM 算法使用其当前的假设 h 代替实际参数θ，以估计 Y 的分布。现定义一个函数 $Q(h'/h)$，在 $\theta=h$ 和全部数据 Y 的观察到的部分 X 的假定之下，它将 $E[\ln P(Y/h')]$作为 h'的一个函数给出。

$$Q(h'/h) = E[\ln P(Y/h') \mid h,X]$$

在 EM 算法的一般形式里，它重复以下两个步骤直至收敛。

第 1 步，估计(E)步骤。使用当前假设 h 和观察到的数据 X 来估计 Y 上的概率分布以计算 $Q(h'/h)$：

$$Q(h'/h) = E[\ln P(Y/h') \mid h,X] \tag{7-9}$$

第 2 步，最大化(M)的步骤。将假设 h 替换为使 Q 函数最大的假设 h'：

$$h \leftarrow \arg\max_{h'} Q(h'/h) \tag{7-10}$$

当函数 Q 连续时，EM 算法收敛到似然函数 $P(Y/h')$的一个不动点。若此似然函数有唯一的最大值，EM 算法可以收敛到这个对 h'的全局极大似然估计。否则，它只能保证收敛到一个局部最大值。因此，EM 算法与其他最优方法有同样的局限性。

7.4 回归分析

“回归”最初是遗传学中的名词,由英国生物学家兼统计学家高尔登首先提出,他在研究人体身高的时候,发现高个子父母的子女身高有低于其父母身高的趋势,而矮个子父母的子女身高有高于其父母身高的趋势,从整体的发展趋势看,高矮个子从两个方向回归于平均人口的平均身高。“回归”这个名词,从此一直为生物学和统计学所沿用。

回归的现代含义和过去大不相同,回归分析是研究变量之间相关关系的一种统计推断法。回归分析,是指在相关分析的基础上,把变量之间的具体变动关系模型化,求出关系方程式,即一个能够反映变量间变化关系的函数关系式,并据此进行估计和推算。通过回归分析,可以将相关变量之间不确定、不规则的数量关系一般化、规范化,从而可以根据自变量的某一个给定值推断出因变量的估计值。

回归分析包括多种类型。根据所涉及变量的多少不同,可分为一元回归和多元回归。一元回归是指两个变量之间的回归,其中一个变量是自变量,另一个变量是因变量。根据变量变化的表现形式不同,回归分析也可分为线性回归和非线性回归。对具有直线相关关系的现象配之以直线方程进行回归分析,即线性回归;对具有曲线相关关系的现象配之以曲线方程进行回归分析,则称为非线性回归。

7.4.1 一元线性回归

一元线性回归将一个随机变量 Y 视为另一个变量 x 的线性函数,即:$Y=a+bx$,其中,a 和 b 是回归系数,分别表示直线在 Y 轴的截距和直线的斜率。

现讨论如何根据观测值(x_i,y_i),$i=1,2,\cdots,n$ 估计回归函数 $f(x)=a+bx$ 中的回归系数。

采用最小二乘法,记平方和

$$Q(a,b)=\sum_{i=1}^{n}(y_i-a-bx_i)^2 \tag{7-11}$$

找使 $Q(a,b)$达到最小的 a、b 作为其估计,即

$$Q(\hat{a},\hat{b})=\min Q(a,b) \tag{7-12}$$

为此,令

$$\begin{cases}\dfrac{\partial Q}{\partial a}=2\sum\limits_{i=1}^{n}(y_i-a-bx_i)=0\\[2ex]\dfrac{\partial Q}{\partial b}=2\sum\limits_{i=1}^{n}(y_i-a-bx_i)x_i=0\end{cases} \tag{7-13}$$

解得

$$\begin{cases}b=\dfrac{L_{xy}}{L_{xx}}\\[2ex]a=\bar{y}-b\bar{x}\end{cases} \tag{7-14}$$

其中,$\bar{x}$ 是 $x_1,x_2,\cdots,x_n$的平均值,而 $\bar{y}$ 是 $y_1,y_2,\cdots,y_n$的平均值,

$$L_{xx}=\sum_{i=1}^{n}(x_i-\overline{x})^2=\sum_{i=1}^{n}x_i^2-\frac{1}{n}\left(\sum_{i=1}^{n}x_i\right)^2$$

$$L_{xy}=\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})=\sum_{i=1}^{n}x_iy_i-\frac{1}{n}\left(\sum_{1}^{n}x_i\right)\left(\sum_{1}^{n}y_i\right) \tag{7-15}$$

例如，某种合成纤维的强度与其拉伸倍数有关。表 7-2 是 24 个纤维样品的强度与相应的拉伸倍数的实测记录。试求这两个变量间的回归表达式。

表 7-2　合成纤维系数表

编　号	1	2	3	4	5	6	7	8	9	10	11	12
拉伸倍数 x	1.9	2.0	2.1	2.5	2.7	2.7	3.5	3.5	4.0	4.0	4.5	4.6
强度 y(Mpa)	1.4	1.3	1.8	2.5	2.8	2.5	3.0	2.7	4.0	3.5	4.2	3.5
编　号	13	14	15	16	17	18	19	20	21	22	23	24
拉伸倍数 x	5.0	5.2	6.0	6.3	6.5	7.1	8.0	8.0	8.9	9.0	9.5	10.0
强度 y(Mpa)	5.5	5.0	5.5	6.4	6.0	5.3	6.5	7.0	8.5	8.0	8.1	8.1

将观察值(x_i,y_i)，$i=1,\cdots,24$ 在平面直角坐标系下用点标出，所得的图称为散点图。从本例的散点图看出，强度 y 与拉伸倍数 x 之间大致呈现线性相关关系，一元线性回归模型是适用于 y 与 x 的。

利用上述公式：

$$n=24$$

$$\sum x_i=127.5,\quad \sum y_i=113.1$$

$$\sum x_i^2=829.61,\quad \sum y_i^2=650.93,\quad \sum x_iy_i=731.6$$

$$L_{xx}=829.61-\frac{1}{24}\times 127.5^2=152.266$$

$$L_{xy}=731.6-\frac{1}{24}\times 127.5\times 113.1=130.756$$

$$L_{yy}=650.93-\frac{1}{24}\times 113.1^2=117.946$$

$$\begin{cases} b=\dfrac{L_{xy}}{L_{xx}} \\ a=\overline{y}-b\overline{x} \end{cases}$$

$$b=\frac{L_{xy}}{L_{xx}}=0.859,\quad a=\overline{y}-b\overline{x}=0.15$$

由此得强度 y 与拉伸倍数 x 之间的经验公式为 $y=0.15+0.859x$。

7.4.2　多元线性回归

在实际问题中，因变量通常不只受到一个自变量的影响，在这种情况下抛开其他因素，只考虑一个因素显然是不合适的，因此有必要研究多个自变量的回归分析。

假设一个随机变量 Y 与 m 个非随机变量 $X_1,X_2,\cdots,X_m$ 之间存在线性相关关系，则它们之间的关系可以用以下线性回归模型来表示：$Y=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_mX_m+e$。其中，$Y$ 是因变量，$X_i(i=1,2,\cdots,m)$是自变量，$\beta_i(i=0,1,2,\cdots,m)$是模型的参数，称为偏相

关系数;e 是随机误差。

回归参数 $\beta_i(i=0,1,2,\cdots,m)$的估计方法还是最小二乘法。根据样本数据$(y,x_{1j},x_{2j},\cdots,x_{mj})$来估计 $\beta_i(i=0,1,2,\cdots,m)$时要使得产生残差的平方和

$$Q=\sum(y_j-\hat{y}_j)^2=\sum[y_j-(\beta_0+\beta_1x_{1j}+\cdots+\beta_mx_{mj})]^2$$

取极小值。为此,对 Q 分别求 $\beta_i(i=0,1,2,\cdots,m)$的偏导数,并令其等于零,由此可以得到 $m+1$ 个方程。

$$\begin{cases}\dfrac{\partial Q}{\partial\beta_0}=-2\sum_{j=1}^{n}[y_j-(\beta_0+\beta_1x_{1j}+\cdots+\beta_mx_{mj})]=0\\ \dfrac{\partial Q}{\partial\beta_1}=-2\sum_{j=1}^{n}[y_j-(\beta_0+\beta_1x_{1j}+\cdots+\beta_mx_{mj})]x_{1j}=0\\ \vdots\\ \dfrac{\partial Q}{\partial\beta_m}=-2\sum_{j=1}^{n}[y_j-(\beta_0+\beta_1x_{1j}+\cdots+\beta_mx_{mj})]x_{mj}=0\end{cases}\tag{7-16}$$

整理后可得方程组

$$\begin{cases}n\beta_0+\sum_{j=1}^{n}x_{1j}\beta_1+\cdots+\sum_{j=1}^{n}x_{mj}=\sum_{j=1}^{n}y_j\\ \sum_{j=1}^{n}x_{1j}\beta_0+\sum_{j=1}^{n}x_{1j}^2\beta_1+\cdots+\sum_{j=1}^{n}x_{1j}x_{mj}\beta_m=\sum_{j=1}^{n}x_{1j}y_j\\ \vdots\\ \sum_{j=1}^{n}x_{mj}\beta_0+\sum_{j=1}^{n}x_{mj}x_{1j}\beta_1+\cdots+\sum_{j=1}^{n}x_{mj}^2\beta_m=\sum_{j=1}^{n}x_{mj}y_j\end{cases}\tag{7-17}$$

对于自变量 $X_1,X_2,\cdots,X_m$ 和因变量 Y 共有 n 组观察数据。x_{ik}表示自变量 X_i 的第 k 次观察值,y_i 表示因变量 Y 的第 i 次观察值。令

$$l_{ij}=\sum_{k=1}^{n}(x_{ik}-\overline{x_i})(x_{jk}-\overline{x_j}),\quad i,j=1,2,\cdots,m$$

$$l_{i0}=\sum_{k=1}^{n}(x_{ik}-\overline{x_i})(y_k-\bar{y}),\quad i=1,2,\cdots,m$$

$$l_{00}=\sum_{k=1}^{n}(y_k-\bar{y})^2\tag{7-18}$$

则回归系数 $\beta_i(i=0,1,2,\cdots,m)$可以由方程组求出:

$$\begin{cases}l_{11}\beta_1+l_{12}\beta_2+\cdots+l_{1m}=l_{10}\\ l_{21}\beta_1+l_{22}\beta_2+\cdots+l_{2m}=l_{20}\\ \vdots\\ l_{m1}\beta_1+l_{m2}\beta_2+\cdots+l_{mm}=l_{m0}\end{cases}$$

$$\text{常数项 }\beta_0=Y-\sum\beta_i\cdot\overline{x_i}\tag{7-19}$$

7.4.3 非线性回归

当判定变量间的关系大致是一条直线时,就可以拟合一条直线反映其变动关系。但是

在很多情况下，变量间的关系呈曲线形式，即非线性的，这时就应拟合一条曲线来反映变量间的关系。

非线性回归的主要模型有以下几种。

(1) 抛物线模型(二次曲线模型)：$y=\beta_0+\beta_1 x+\beta_2 x^2+\varepsilon$

(2) 双曲线模型：$y=\beta_1+\beta_2 \dfrac{1}{x}+\varepsilon$

(3) 对数模型：$y=\beta_1+\beta_2 \ln x+\varepsilon$

(4) 三角函数模型：$y=\beta_1+\beta_2 \sin x+\varepsilon$

(5) 指数函数模型：$y=\beta_0 e^{\beta x+\varepsilon_i}$

(6) 幂函数模型：$y=\alpha x_j^b+\varepsilon_i$

(7) 多项式模型：$y=\beta_0+\beta_1 x+\beta_2 x^2+\cdots+\beta_p x^p+\varepsilon$

(8) 修正指数增长模型：$y=\alpha+\beta r^{x_i}+\varepsilon_i$

根据非线性回归模型线性化的不同性质，这些模型可以分为 3 种类型。

1. 直接换元法

这类非线性回归模型通过简单的变量换元可以直接化为线性回归模型，如双曲线模型、二次曲线模型、对数模型、三角模型。由于这类模型的因变量没有变形，所以可以直接采用最小平方法估计回归系数并进行检验和预测。

2. 间接代换法

这类非线性回归模型经常通过对数变形代换间接地化为线性回归模型，如指数模型、幂函数模型。由于这类模型在对数变形代换过程中改变了因变量的形态，使得变形后模型的最小平方估计失去了原模型的残差平方和最小的性质，从而估计不到原模型的最佳回归系数，造成了回归模型与原数列之间的较大偏差。

3. 非线性型

这类非线性回归模型属于不可线性化的非线性回归模型，如修正指数增长模型。

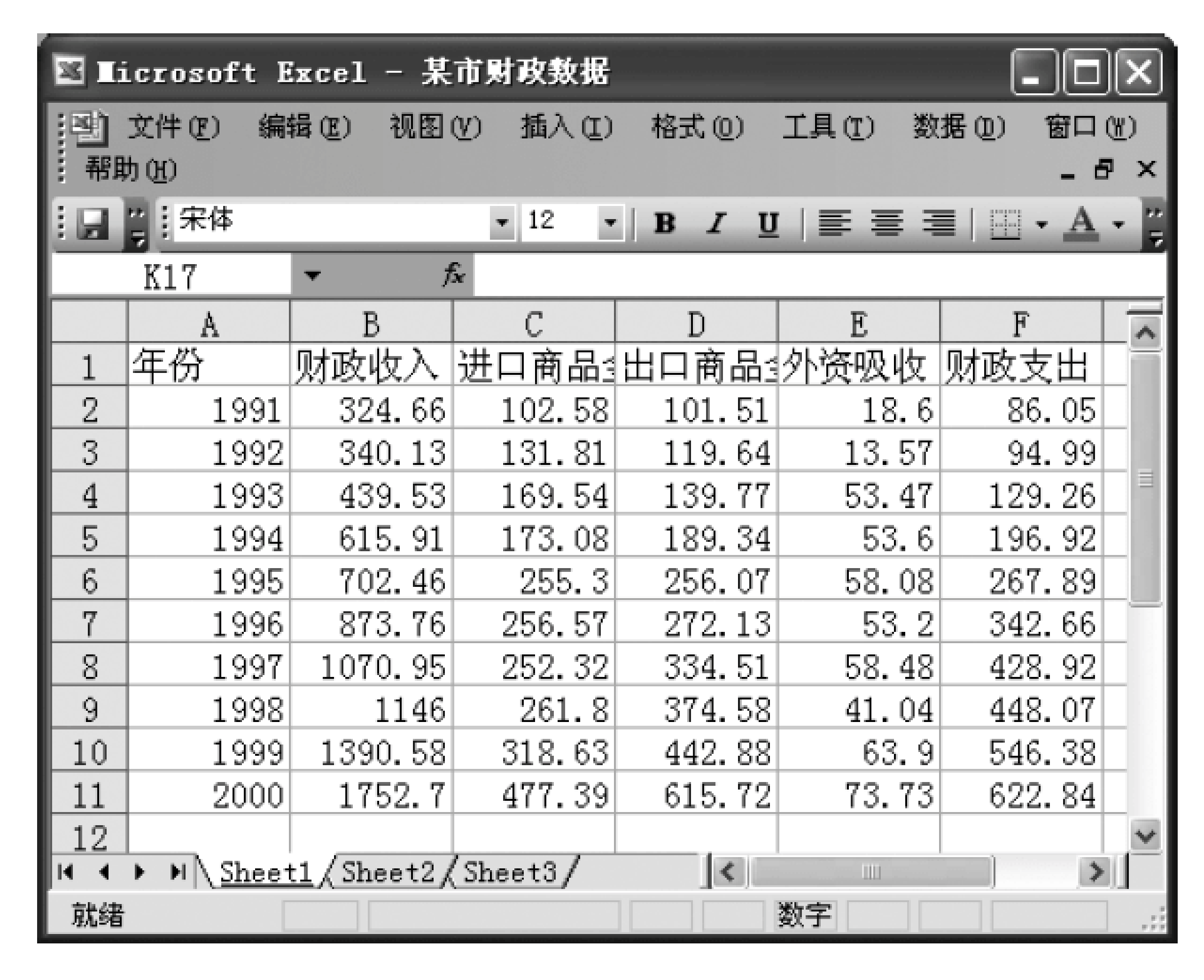
7.5 利用 SQL Server 2005 进行线性回归分析

图 7-4 为某市 1991—2000 年的财政数据，数据包括每年的财政收入、进口商品金额、出口商品金额、外资吸收金额和财政支出。问题是如何利用数据挖掘技术确定财政收入和进口商品金额、出口商品金额、外资吸收金额、财政支出之间的关系。

将每两个指标下的数据在平面直角坐标系下用点标出，可得到指标之间的矩阵散点图。通过矩阵散点图 7-5，可以看出财政收入分别和进口商品金额、出口商品金额、外资吸收金额、财政支出之间呈近似的线性关系。下面使用 SQL Server 2005 软件，建立财政收入和进口商品金额、出口商品金额、外资吸收金额、财政支出之间的多元线性回归模型。

数据挖掘模型的创建由以下步骤完成。

(1) 右击项目数据挖掘向导下的“挖掘结构”，选择“新建挖掘结构”，打开“数据挖掘向导”对话框，单击“下一步”按钮，弹出“选择定义方法”对话框，单击“下一步”按钮，进行下一步操作。



	A	B	C	D	E	F
1	年份	财政收入	进口商品	出口商品	外资吸收	财政支出
2	1991	324.66	102.58	101.51	18.6	86.05
3	1992	340.13	131.81	119.64	13.57	94.99
4	1993	439.53	169.54	139.77	53.47	129.26
5	1994	615.91	173.08	189.34	53.6	196.92
6	1995	702.46	255.3	256.07	58.08	267.89
7	1996	873.76	256.57	272.13	53.2	342.66
8	1997	1070.95	252.32	334.51	58.48	428.92
9	1998	1146	261.8	374.58	41.04	448.07
10	1999	1390.58	318.63	442.88	63.9	546.38
11	2000	1752.7	477.39	615.72	73.73	622.84
12						

图 7-4　某市 10年财政数据

图 7-5　矩阵散点图

(2) 在如图 7-6 所示的下拉列表框中选取“Microsoft 线性回归”选项,单击“下一步”按钮,进行下一步操作。

(3) 如图 7-7 所示,在“选择数据源视图”对话框的“可用数据源视图”列表中显示了创建的某市财政数据源视图,选中该视图选项,单击“下一步”按钮,进行下一步操作。

(4) 如图 7-8 所示,在“指定表类型”对话框中可以看到某市财政数据源视图包含的数据表,在各个选项右边勾选不同的复选框,其中进口商品金额、出口商品金额、外资吸收、财政支出作为输入,年份作为主键,财政收入作为可预测变量,单击“下一步”按钮,进行下一步操作。

(5) 如图 7-9 所示,在“指定列的内容和数据类型”对话框中显示了指定“年份”的内容类型为 Key,其余列内容类型均为 Continuous;所有列的数据类型均为 Double,单击“下一步”按钮,进行下一步操作。

(6) 如图 7-10 所示,在“完成向导”页面中完成挖掘结构的创建。

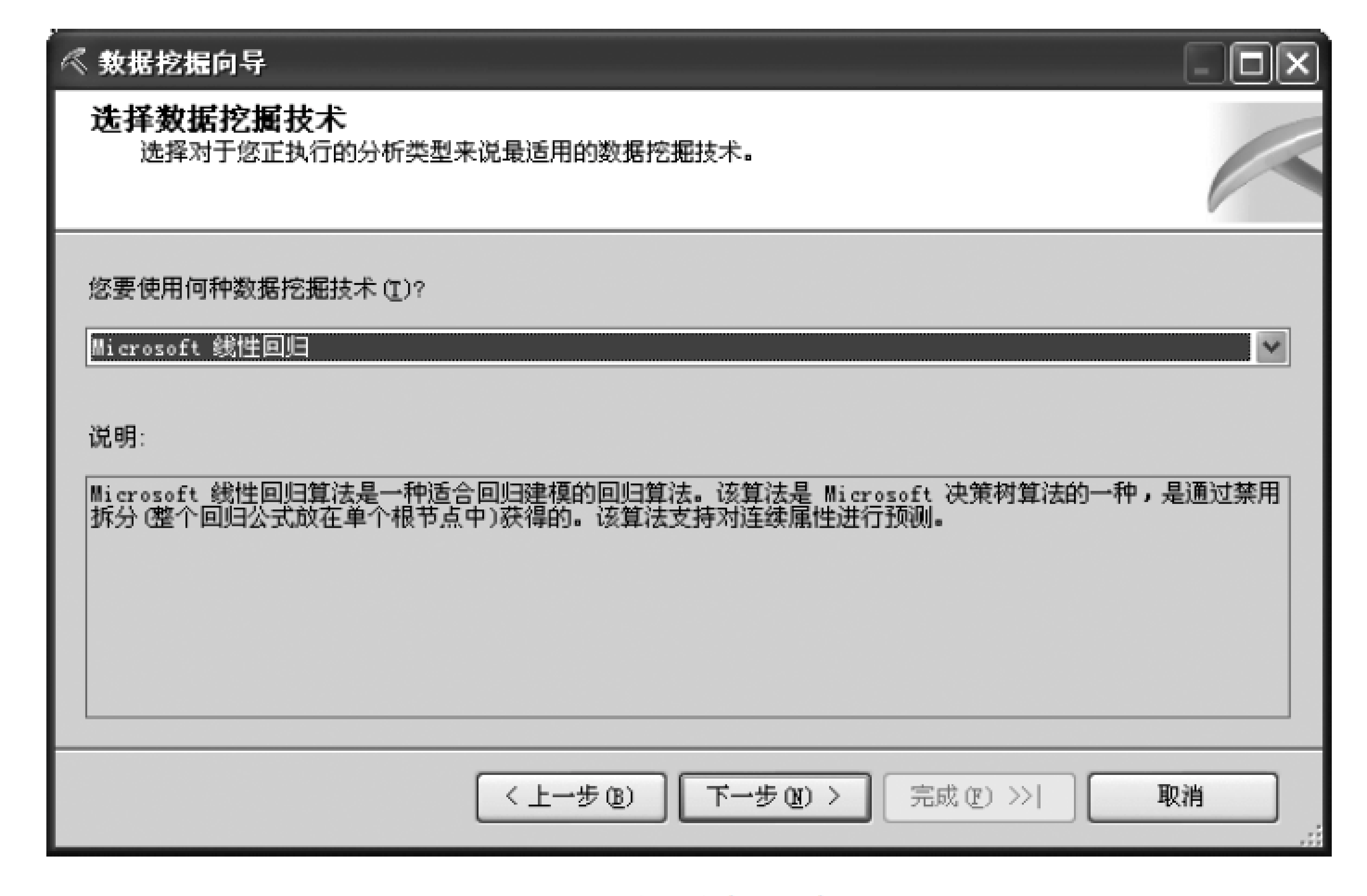


图 7-6　选择数据挖掘技术



图 7-7　选择数据源视图



图 7-8　指定表类型



图 7-9　指定列的内容和数据类型



图 7-10　完成数据挖掘结构的创建

(7) 单击“挖掘模型”选项卡下的“依赖关系网络”，可以清晰地看到进口商品金额、出口商品金额、外资吸收、财政支出这 4 个输入和可预测变量财政收入之间的关系，其结果如图 7-11 所示。

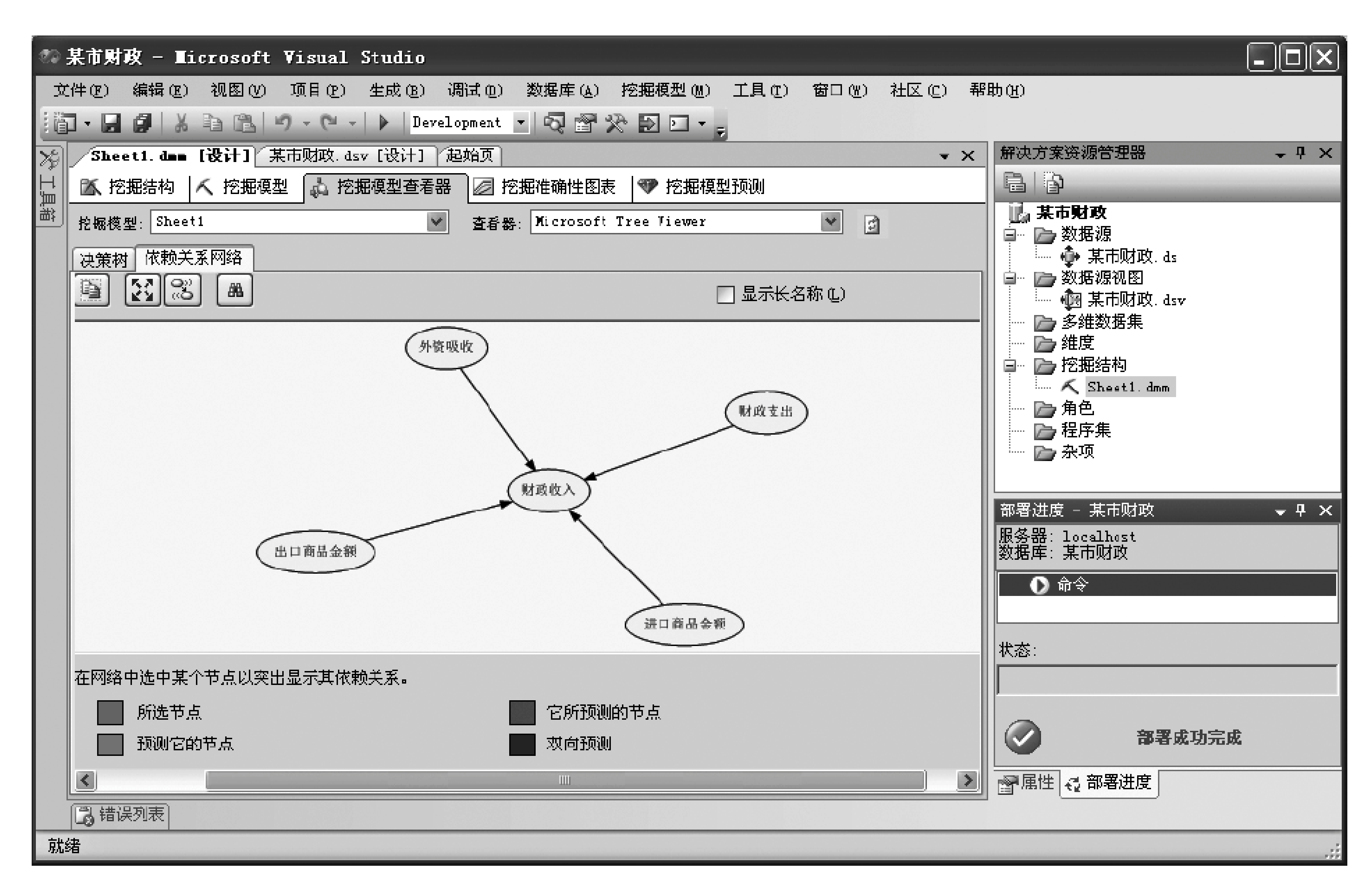


图 7-11　依赖关系网络

(8) 单击"挖掘准确性图表"选项卡下的"提升图"和"分类矩阵",其结果如图 7-12 所示。

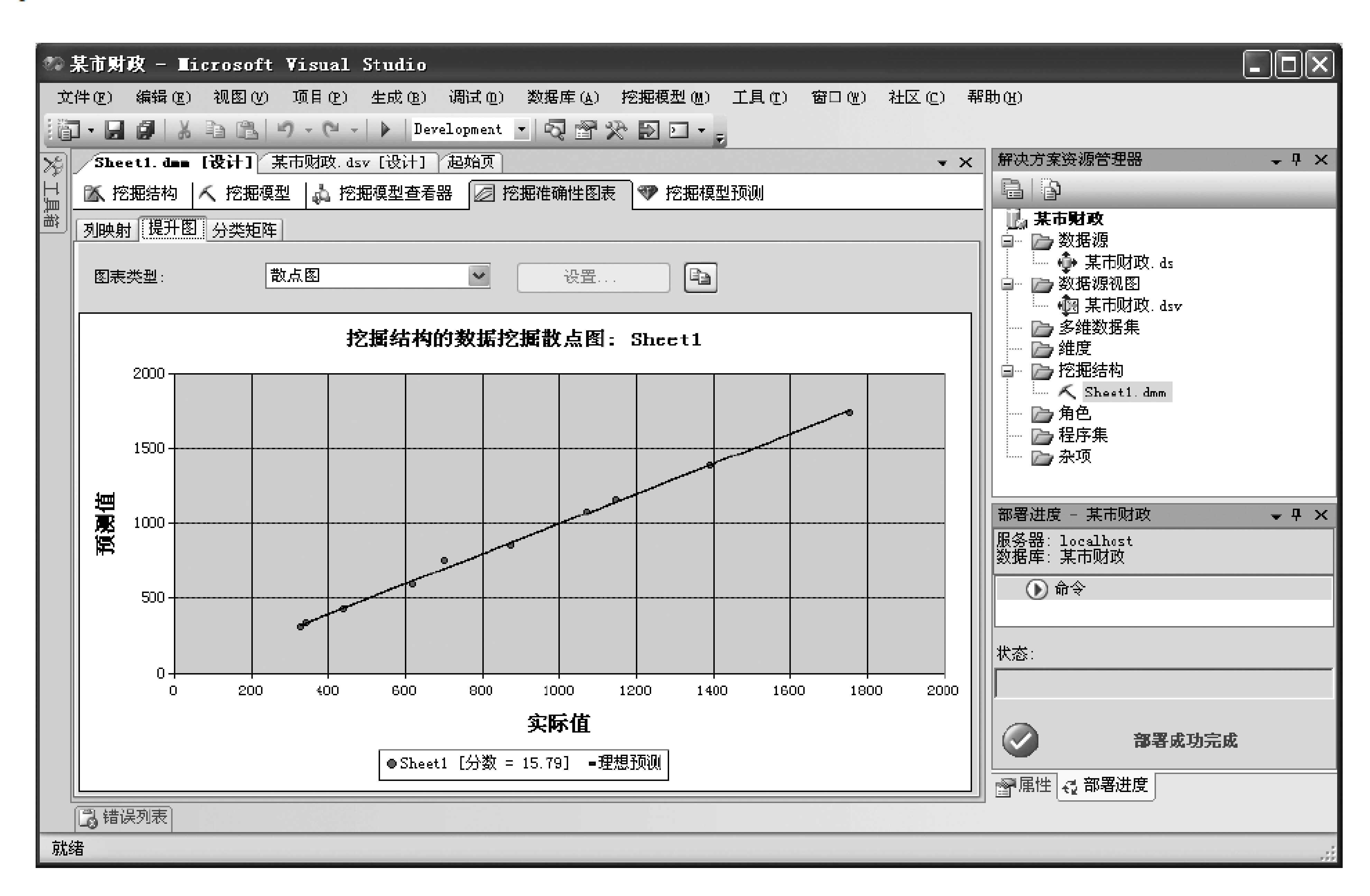


图 7-12　提升图

图 7-12 中显示了预测值和实际值之间的拟合关系,可以看出在建立回归模型的基础上预测值和实际值基本拟合。

(9) 将图 7-12 右下部署进度复选框拖出,如图 7-13 所示。由此图可以得到财政收入和进口商品金额、出口商品金额、外资吸收金额、财政支出之间的多元线性回归方程。

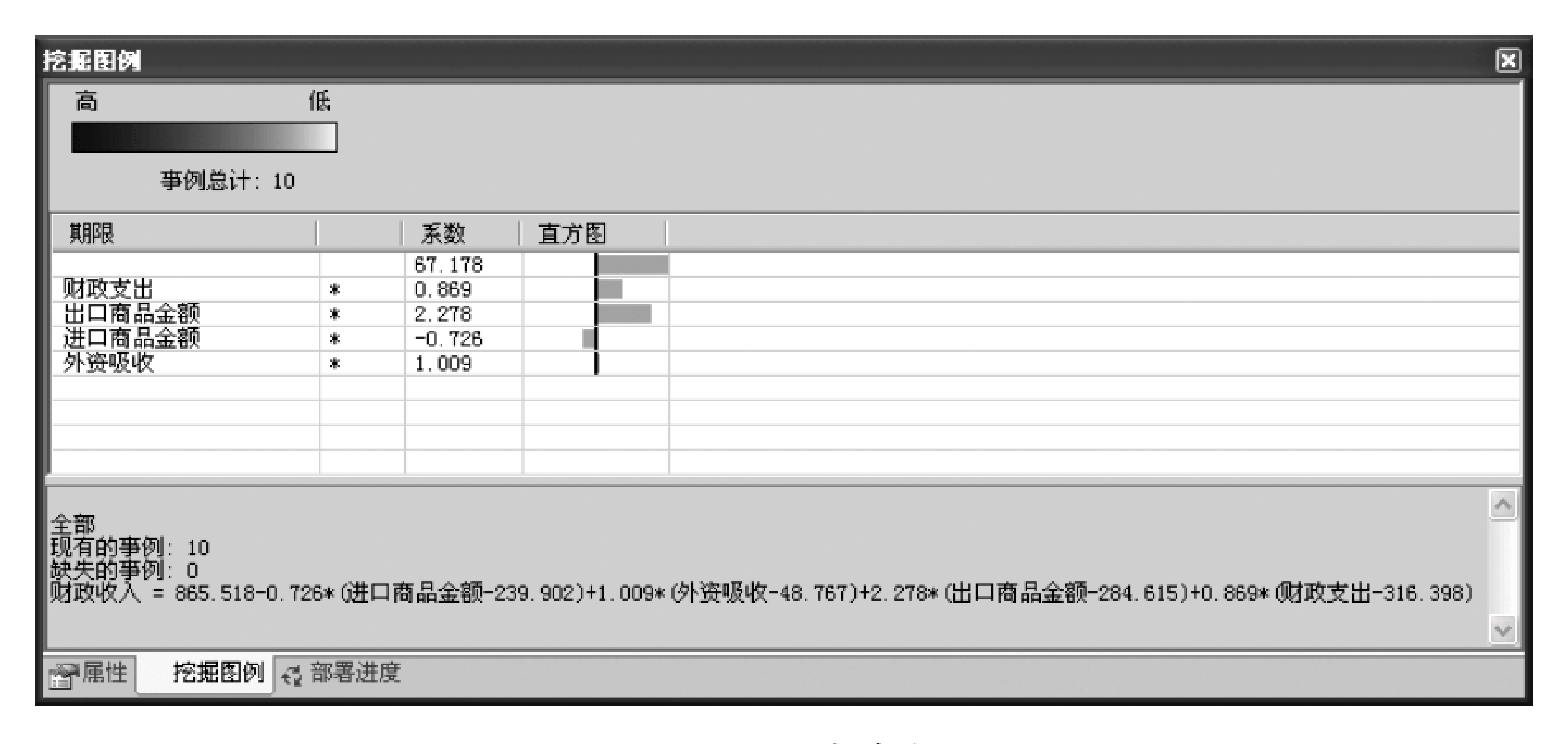


图 7-13　回归方程

小结

基于统计的学习方法在数据挖掘中占有重要位置。本章较为具体地介绍了数据挖掘中常用的统计学习方法,包括朴素贝叶斯分类、贝叶斯信念网络、EM 算法和回归分析方法。最后详细介绍了基于 SQL Server 2005 的线性回归实现技术,使读者进一步体会如何利用 SQL Server 2005 具体实现线性回归挖掘。

习题 7

1. 什么是贝叶斯定理?
2. 简述如何利用朴素贝叶斯方法进行分类。
3. 简述贝叶斯信念网络的特点及其应用。
4. 简述 EM 算法的基本思想。
5. 简述线性回归的思想。
6. 非线性回归的模型有哪些?

第 8 章　人工神经网络方法

从数学和物理方法以及信息处理的角度对人脑神经网络进行抽象，并建立某种简化模型，称为人工神经网络（artificial neural network，ANN）。在模式识别、系统辨识、信号处理、自动控制、组合优化、预测预估、故障诊断、数据挖掘、医学和经济学等领域，人工神经网络已经成功解决了许多现代计算机难以解决的实际问题，表现出良好的智能特性和潜在的应用前景。

人工神经网络的特点和优势主要表现在以下 3 个方面。第一，具有自学习功能。例如实现图像识别时，只要先把不同的图像样本和对应的识别结果输入人工神经网络，网络就会通过自学习功能，学习识别类似的图像。第二，具有联想存储功能。人工神经网络的反馈网络可以实现这种联想，例如，经过训练的神经网络可以从“眼睛”特征恢复整个人脸图像，这叫做自联想，从“勺子”联系出“筷子”、“碗”等，这叫做互联想。第三，具有高速寻找优化解的能力。寻找某个复杂问题的优化解往往需要很大的计算量，利用一个针对特定问题而设计的反馈型人工神经网络，发挥计算机的高速运算能力，可以很快找到优化解。

由于人工神经网络具有高度的抗干扰能力和可以对未训练数据进行分类等优点，在预测型知识挖掘中，它已经成为很有用的一种模式结构。

8.1　人工神经网络的基本概念

8.1.1　人工神经元原理

神经生理学家和神经解剖学家早已证明，人的思维是通过人脑完成的，神经元是组成人脑的最基本单元，人脑神经元大约有 10^{11}～10^{12} 个。

神经元由细胞体、树突和轴突三部分组成，是一种根须状的蔓延物。神经元的中心有一闭点，称为细胞体，它能对接收到的信息进行处理。细胞体周围的纤维有两类，轴突是较长的神经纤维，是发出信息的。树突的神经纤维较短，而分支很多，是接收信息的。一个神经元的轴突末端与另一个神经元的树突之间密切接触，传递神经元冲动的地方称为突触。经过突触的冲动传递是有方向性的，不同的突触进行的冲动传递效果不一样，有的使后一神经元发生兴奋，有的使它受到抑制，每个神经元可有 10～10^4 个突触。这表明大脑是一个广泛连接的复杂网络系统。从信息处理功能看，神经元具有如下性质：

（1）多输入，单输出；

（2）突触兼有兴奋和抑制两种性能；

（3）可时间加权和空间加权；

（4）可产生脉冲；

（5）脉冲传递；

（6）非线性（有阈值）。

神经元的数学模型可用图 8-1 表示。

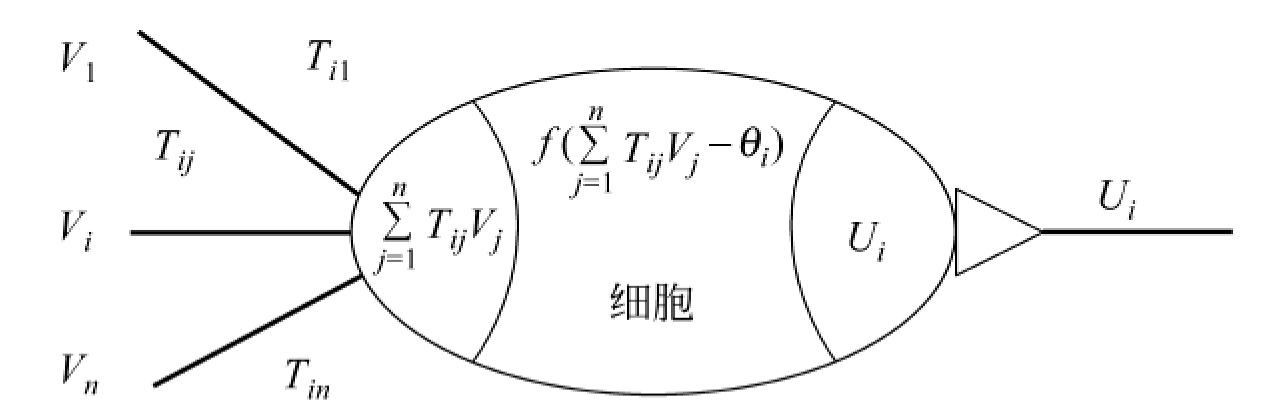


图 8-1　神经元模型

该神经元模型输入输出关系为

$$S_i = \sum_{j=1}^{n} T_{ij} V_j - \theta_i$$

$$U_i = f(S_i) \tag{8-1}$$

其中，$V_1, V_2, \cdots, V_n$ 为输入；U_i 为神经元的输出；T_{ij} 为外面神经元与该神经元的连接强度（即权）；θ_i 为阈值；$f(x)$ 为该神经元的作用函数。图 8-2 表示了几种常见的作用函数。

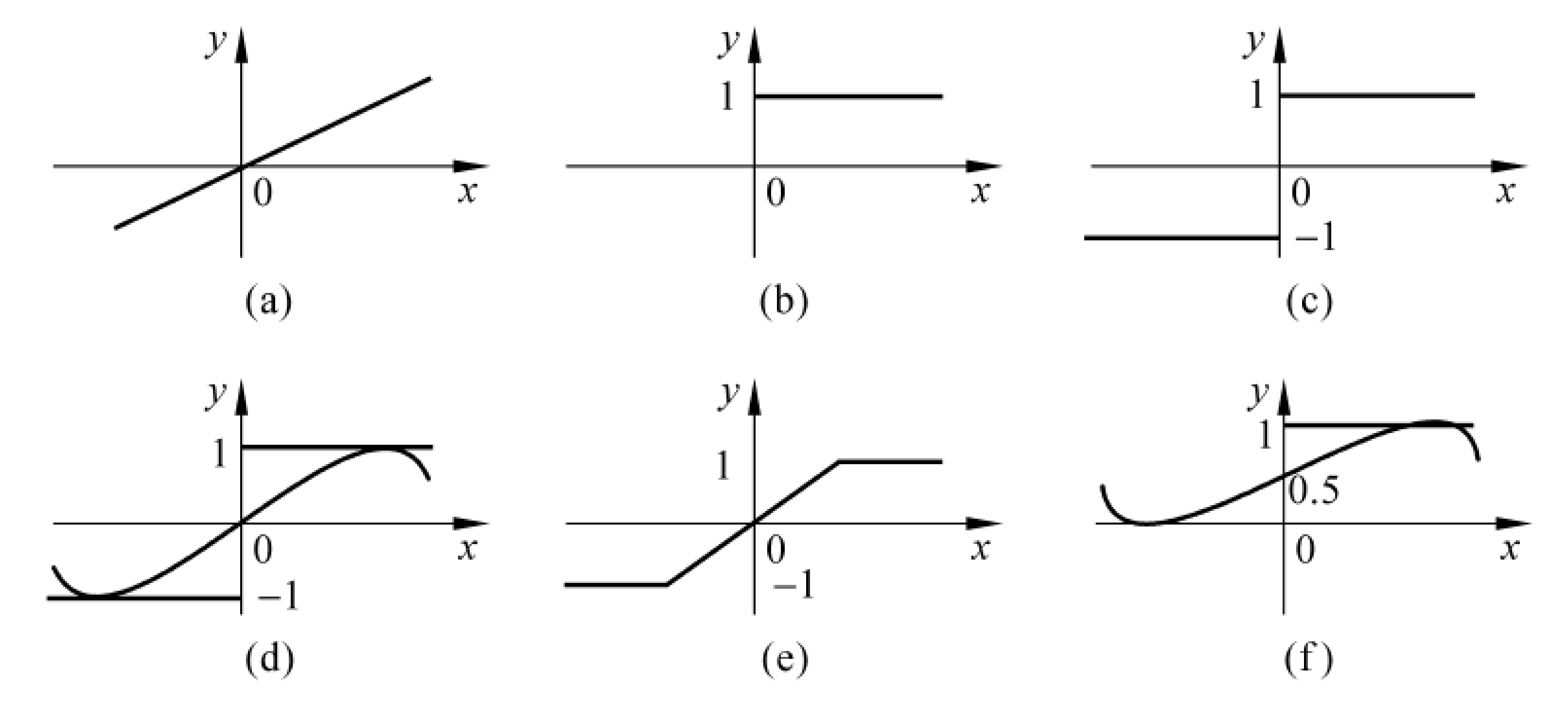


图 8-2　常见的作用函数

在图 8-2 中，各作用函数的解析表达式如下。

(a) 比例函数：

$$y = f(x) = x \tag{8-2}$$

(b) [0,1]阶跃函数：

$$y = f(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leqslant 0 \end{cases} \tag{8-3}$$

(c) [−1,1]符号函数：

$$y = f(x) = \begin{cases} 1, & x > 0 \\ -1, & x \leqslant 0 \end{cases} \tag{8-4}$$

(d) (−1,1)双曲函数：

$$y = f(x) = \frac{1 - \mathrm{e}^{-x}}{1 + \mathrm{e}^{-x}} \tag{8-5}$$

(e) 饱和函数：

$$y=f(x)=\begin{cases}1, & x\geqslant \dfrac{1}{k}\\ kx, & -\dfrac{1}{k}\leqslant x<\dfrac{1}{k}\\ -1, & x<-\dfrac{1}{k}\end{cases}\tag{8-6}$$

(f) (0,1)S 型函数：

$$y=f(x)=\frac{1}{1+e^{-x}}\tag{8-7}$$

8.1.2 人工神经网络拓扑结构

人工神经网络是一个并行和分布式的信息处理网络结构，该网络结构一般由许多个神经元组成，每个神经元有一个单一的输出，它可以连接到很多其他的神经元，其输入有多个连接通路，每个连接通路对应一个连接权系数。

严格来说，神经网络是一个具有如下性质的有向图。

(1) 对于每个结点有一个状态变量 V_j；

(2) 结点 j 到结点 i 有一个连接权系数 T_{ij}；

(3) 对于每个结点有一个阈值 θ_i；

(4) 对于每个结点定义一个变换函数(作用函数) $f(x)$。

图 8-3 表示了两种典型的神经网络结构，图 8-3(a)为前馈型网络，图 8-3(b)为反馈型网络。

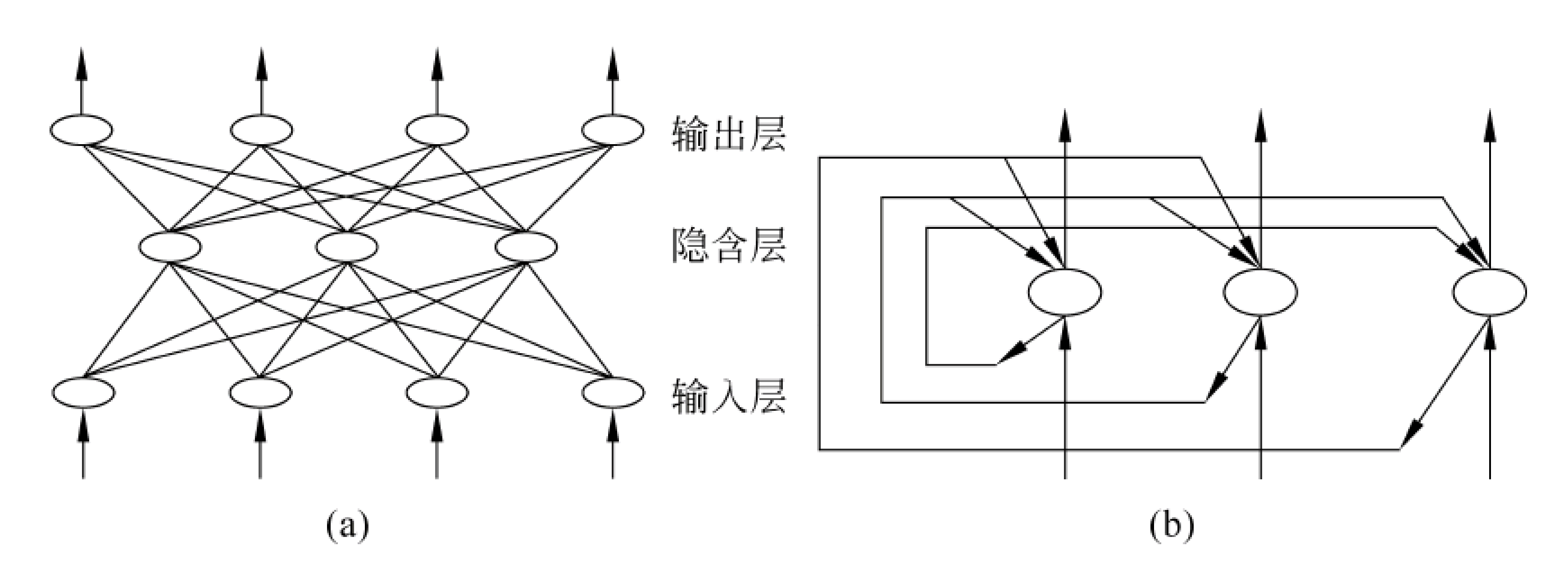


图 8-3 典型的神经网络结构

人工神经网络是生物神经网络的一种模拟和近似。它主要从两个方面进行模拟：一种是从结构和实现机理方面进行模拟，它涉及生物学、生理学、心理学、物理及化学等许多基础学科；另一种是从功能上加以模拟，即尽量使得人工神经网络具有生物神经网络的某些功能特性，如学习、识别、控制等功能。本章着重于后者，主要介绍几种典型的神经网络模型，并讨论它们在数据挖掘中的应用。

8.1.3 人工神经网络学习算法

神经网络之所以能够胜任一些复杂的工作是因为它有学习的能力，神经网络的工作过程可分为两个阶段。首先对神经网络进行训练(学习期)，在学习期主要是利用给定的数据或知识来调整网络的各种参数(连接权、阈值，甚至还包括网络拓扑结构)。当学习结束后，

神经网络便学到了所要求的知识。在下一个阶段(工作期)神经网络就利用学习期所学到的知识,对网络输入做出正确的响应。

1. 神经网络的学习方式

神经网络的学习方式有 3 种:监督学习(有导师学习)、非监督学习(无导师学习)和再励学习(强化学习)。

(1) 监督学习。监督学习需要有一组给定的样本(输入输出数据对),其中包含了输入数据和对应的正确输出,神经网络就利用样本和自身的输出间的误差不断调整连接强度参数,直到神经网络的输出和正确的输出接近到某一程度为止,这样神经网络就模拟了输入空间到输出空间的映射关系。

(2) 非监督学习。进行非监督学习时,外部数据没有提供正确的输出,只是根据外部数据的某些统计规律来调整连接强度参数或拓扑结构,其实这是一种自组织的过程。这样可以表示出外部数据的某些固有特征,如聚类或统计分布特征。

(3) 再励学习。再励学习处于以上两种学习之间,既不给出正确答案又不是什么参考都没有,而是对神经网络的输出给出评价信息,通过奖惩来完善神经网络的权值,从而改善网络性能。

2. 神经网络的学习规则

(1) 误差纠正学习。令 $y_k(n)$ 为输入是 $x_k(n)$ 时的神经元 k 在 n 时刻的实际输出,$d_k(n)$ 表示应有的输出,则误差信号可写为

$$e_k(n) = d_k(n) - y_k(n) \tag{8-8}$$

误差纠正学习的最终目的是使某一基于 $e_k(n)$ 的目标函数达到最小,以使网络中每一输出单元的实际输出在某种统计意义上逼近应有输出。一旦选定了目标函数形式,误差纠正学习就变成了典型的最优化问题,通常用的目标函数是均方误差判据,定义为误差平方和的均值 J:

$$J = E\left[\frac{1}{2}\sum_k e_k^2(n)\right] \tag{8-9}$$

其中 E 为求期望算子,上式的前提是被学习的过程是宽平稳的,具体方法可用最优梯度下降法。直接用 J 在时刻 n 的瞬时值 $\zeta(n)$ 代替 J,即

$$\zeta(n) = \frac{1}{2}\sum_k e_k^2(n) \tag{8-10}$$

问题变为求 $\zeta(n)$ 对权值 w 的极小值,根据梯度下降法可得

$$\Delta w_{kj} = \eta e_k(n) x_j(n) \tag{8-11}$$

其中 η 为学习步长,这就是通常所说的误差纠正学习,又叫 delta 学习规则。

(2) Hebb 学习。Hebb 学习是由神经心理学家 Hebb 提出的学习规则,可归纳为“当某一连接两端的神经元同步激发或同步抑制时该连接强度应增强,反之减弱”。

Hebb 学习规则为,若 i 与 j 两个神经元之间同时处于兴奋状态,则它们之间的连接应加强,即

$$\Delta \bar{\omega}_{ij} = \alpha S_i S_j, \quad \alpha > 0 \tag{8-12}$$

这一规则与“条件反射”学说一致,并得到神经细胞学说的证实。设 $\alpha=1$,当 $S_i=S_j=1$

时，$\Delta\bar{\omega}_{ij}=1$，在 S_i，S_j 中有一个为 0 时，$\Delta\bar{\omega}_{ij}=0$。

（3）竞争（competitive）学习。在竞争学习时，网络各输出单元互相竞争，最后达到只有一个最强者激活，最常见的情况是输出神经元之间有侧向抑制性连接，这样原来输出单元中如有某一单元较强，则它将获胜并抑制其他单元，最后只有此强者处于激活状态。

神经网络有很多种学习算法，一个神经网络要选用什么算法与网络结构规模和实际问题的性质有关。常见的学习算法有以下几种：误差反向传播（error back propagation，BP）学习算法、遗传算法、最小二乘学习算法、随机梯度法、模拟退火算法、卡尔曼滤波器算法等。下面将结合具体的神经网络详细介绍相应的学习算法。

8.1.4 人工神经网络泛化

建立神经网络模型的一个重要目标是通过对已知环境信息的学习，掌握其中的规律，从而对新的环境信息做出正确的预测，这个目标是通过神经网络模型的泛化（generalization）能力来体现的。泛化能力定义为：经训练（学习）后的预测模型对未在训练集中出现（但具有统一规律性）的样本做出正确反映的能力。学习不是简单地记忆已经学过的输入，而是通过对有限个训练样本的学习，得到隐含在样本中的有关环境本身的内在规律性。例如，对有导师学习的网络，通过对已有样本的学习，将所提取的样本中的非线性映射关系存储在权值矩阵中，在其后的工作阶段，当向网络输入训练时未曾见过的非样本数据（与训练集同分布）时，网络也能完成由输入空间向输出空间的正确映射。

神经网络的泛化能力与其预测功能密切相关，一般来说，神经网络模型的泛化能力取决于三个主要因素，即问题本身的复杂程度、网络权值参数的初值以及样本量的大小。问题复杂，学习样本数量少，网络权值初始化不理想，则泛化能力弱；反之泛化能力强。具有较好的泛化能力是神经网络设计的评价指标之一，有许多不同的度量方法，有兴趣的读者可以参阅有关神经网络设计方法的书籍。

8.2 误差反向传播（BP）神经网络

BP 模型是 1986 年由 Rumelhart 和 McCelland 领导的科学家小组提出的，具有如图 8-3(a) 所示的分层结构，最下面一层是输入层，中间是隐含层，最上面一层是输出层。其信息从输入层依次向上传递，直至输出层。该网络实际是一种多层感知器（multi perceptron）前馈网络，由于连接权的调整采用的是误差反向传播的学习算法，因此也称为 BP 网络。该网络是至今为止应用最广泛的神经网络。

8.2.1 BP 神经网络的拓扑结构

BP 神经网络不仅有输入结点、输出结点，而且还有一层或多层隐含结点，神经元的变换函数采用(0,1)S 型函数。如图 8-4 所示。

图 8-4 BP 神经网络的拓扑结构

图 8-4 所示的网络中，最下面的层为输入层，第 Q 层为输出层，中间各层为隐含层，设第 q 层（$q=1$，$2,\cdots,Q$）的神经元个数为 n_q，输入到第 q 层的第 i 个

神经元的连接权系数为 $\bar{\omega}_{ij}^{(q)}$ ($i=1,2,\cdots,n_q$; $j=1,2,\cdots,n_{q-1}$)。该网络的输入输出变换关系为：

$$s_i^{(q)} = \sum_{j=0}^{n_q-1} \bar{\omega}_{ij}^{(q)} x_j^{(q-1)} \quad x_0^{(q-1)} = \theta_i^{(q)}, \quad \bar{\omega}_{i0}^{(q)} = -1$$

$$x_i^{(q)} = f(s_i^{(q)}) = \frac{1}{1+\mathrm{e}^{-\mu s_i^{(q)}}}$$

$$i = 1,2,\cdots,n_q \quad j = 1,2,\cdots,n_{q-1} \quad q = 1,2,\cdots,Q \tag{8-13}$$

设给定 P 组输入输出样本 $x_p^{(0)} = [x_{p1}^{(0)} x_{p2}^{(0)} \cdots x_{pn_0}^{(0)}]$，$d_p = [d_{p1} d_{p2} \cdots d_{pn_Q}]^T$ ($p=1,2,\cdots,P$) 利用该样本集首先对 BP 网络进行训练，训练的目的是对网络的连接权系数进行学习和调整，以使该网络实现给定的输入输出关系。经过训练的 BP 网络，对于不是样本集中的输入也能给出合适的输出，也就是泛化功能。从函数拟合的角度，它说明 BP 网络具有插值的功能。

8.2.2 BP 神经网络学习算法

误差反向传播(BP)学习算法是实现函数逼近的一种方法。其基本思想是，学习过程由信号的正向传播与误差的反向传播两个过程组成。正向传播时，输入样本从输入层传入，经各隐层逐层处理后，传向输出层。若输出层的实际输出与期望的输出(教师信号)不符，则转入误差的反向传播阶段。误差反传是将输出误差以某种形式通过隐层向输入层逐层反传，并将误差分摊给各层的所有结点，从而获得各层结点的误差信号，此误差信号即作为修正各结点权值的依据。这种信号正向传播与误差反向传播的各层权值调整过程是周而复始地进行的。权值不断调整的过程，也就是网络的学习训练过程。此过程一直进行到网络输出的误差减少到可接受的程度，或进行到预先设定的学习次数为止。

在 BP 神经网络中，输入信号是从输入层到隐层再到输出层传递的。最后一个隐层与输出层之间的连接权是输出误差的显函数，而其他层之间的连接权则是输出误差的隐函数。如果神经元的作用函数是连续可微的，那么每一连接权对输出误差的影响都可以由误差对权值的偏导数定量的描述。此时如果把权值按照梯度的反方向修正则可以使误差减小。这种思想便是 BP 算法的本质。详细计算方法如下：

设取拟合误差的代价函数为

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{i=1}^{n_Q}(d_{pi} - x_{pi}^{(Q)})^2 = \sum_{p=1}^{P} E_p \tag{8-14}$$

即

$$E_p = \frac{1}{2}\sum_{i=1}^{n_Q}(d_{pi} - x_{pi}^{(Q)})^2 \tag{8-15}$$

问题是如何调整连接权系数以使代价函数最小。优化计算的方法很多，比较典型的是采用一阶梯度法，即最速下降法。该方法的关键是计算优化目标函数 E(即上述的误差代价函数)对寻优参数的一阶导数。依次从输出层开始计算如下：

$$\frac{\partial E}{\partial \bar{\omega}_{ij}^{(q)}}, \quad q = Q, Q-1, \cdots, 1$$

由于

$$\frac{\partial E}{\partial \bar{\omega}_{ij}^{(q)}} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \bar{\omega}_{ij}^{(q)}}$$

所以应着重讨论$\frac{\partial E_p}{\partial \bar{\omega}_{ij}^{(q)}}$的计算。

对于第 Q 层有

$$\frac{\partial E_p}{\partial \bar{\omega}_{ij}^{(Q)}} = \frac{\partial E_p}{\partial x_{pi}^{(Q)}} \frac{\partial x_{ij}^{(Q)}}{\partial s_{pi}^{(Q)}} \frac{\partial s_{ipi}^{(Q)}}{\partial \bar{\omega}_{ij}^{(Q)}} = -(d_{pi} - x_{pi}^{(Q)}) f'(s_{pi}^{(Q)}) x_{pi}^{(Q-1)} = -\delta_{pi}^{(Q)} x_{pi}^{(Q-1)} \tag{8-16}$$

其中

$$\delta_{pi}^{(Q)} = -\frac{\partial E_p}{\partial s_{pi}^{(Q)}} = (d_{pi} - x_{pi}^{(Q)}) f'(s_{pi}^{(Q)})$$

$x_{pi}^{(Q)}$,$s_{pi}^{(Q)}$及$x_{pi}^{(Q-1)}$表示利用第 p 组输入样本所算得的结果。

对于第 $Q-1$ 层有

$$\begin{aligned}\frac{\partial E_p}{\partial \bar{\omega}_{ij}^{(Q-1)}} &= \frac{\partial E_p}{\partial x_{pi}^{(Q-1)}} \frac{\partial x_{ipi}^{(Q-1)}}{\partial \bar{\omega}_{ij}^{(Q-1)}} = \left(\sum_{k=1}^{n_Q} \frac{\partial E_p}{\partial s_{pk}^{(Q)}} \frac{\partial s_{pk}^{(Q)}}{\partial x_{pi}^{(Q-1)}} \right) \frac{\partial x_{pi}^{(Q-1)}}{\partial s_{pi}^{(Q-1)}} \frac{\partial s_{pi}^{(Q-1)}}{\partial \bar{\omega}_{ij}^{(Q-1)}} \\ &= \left(\sum_{k=1}^{n_Q} \delta_{pk}^{(Q)} \bar{\omega}_{ki}^{(Q)} \right) f'(s_{ipi}^{(Q-1)}) x_{pj}^{(Q-2)} = -\delta_{pi}^{(Q-1)} x_{pj}^{(Q-2)}\end{aligned} \tag{8-17}$$

其中

$$\delta_{pi}^{(Q-1)} = -\frac{\partial E_p}{\partial s_{pi}^{(Q-1)}} = \left(\sum_{k=1}^{n_Q} \delta_{pk}^{(Q)} \bar{\omega}_{ki}^{(Q)} \right) f'(s_{ipi}^{(Q-1)})$$

显然,它是反向递推计算的公式,即首先计算出 $\delta_{pk}^{(Q)}$ 然后再由上式递推计算出 $\delta_{pi}^{(Q-1)}$,依次类推,可继续反向递推计算出 $\delta_{pi}^{(q)}$ 和$\frac{\partial E_p}{\partial \bar{\omega}_{ij}^{(q)}}$($q=Q-2,\cdots,1$)。从上式看出,在 $\delta_{pi}^{(q)}$ 的表达式中包含了导数项 $f'(s_{pi}^{(q)})$,由于 BP 网络使用 S 形函数,所以其导数可求得如下:

$$x_{pi}^{(q)} = f(s_{pi}^{(q)}) = \frac{1}{1+\mathrm{e}^{-\mu s_{pi}^{(q)}}}$$

$$f'(s_{pi}^{(q)}) = \frac{\mu \mathrm{e}^{-\mu s_{pi}^{(q)}}}{(1+\mathrm{e}^{-\mu s_{pi}^{(q)}})^2} = \mu f(s_{pi}^{(q)})[1 - f(s_{pi}^{(q)})] = \mu x_{pi}^{(q)} (1 - x_{pi}^{(q)}) \tag{8-18}$$

最后可归纳出 BP 网络的学习算法如下:

$$\begin{aligned}&W_{ij}^{(q)}(k+1) = \bar{\omega}_{ij}^{(q)}(k) + \alpha D_{ij}^{(q)}(k+1), \quad \alpha > 0 \\ &D_{ij}^{(q)} = \sum_{p=1}^{P} \delta_{pi}^{(q)} x_{pj}^{(q-1)} \\ &\delta_{pi}^{(q)} = \left(\sum_{k=1}^{n_q+1} \delta_{pk}^{(q+1)} \bar{\omega}_{ki}^{(q+1)} \right) \mu x_{pi}^{(q)} (1 - x_{pi}^{(q)}) \\ &\delta_{pi}^{(Q)} = (d_{pi} - x_{pi}^{(Q)}) \mu x_{pi}^{(Q)} (1 - x_{pi}^{(Q)}) \\ &q = Q, Q-1, \cdots, 1; i = 1, 2, \cdots, n_q; j = 1, 2, \cdots, n_{q-1}\end{aligned} \tag{8-19}$$

该网络实质上是对任意非线性映射关系的一种逼近,由于采用的是全局逼近的方法,因而 BP 网络具有较好的泛化能力。从以上的讨论看出,对于给定的样本集,目标函数 E 是全体连接权系数 $\bar{\omega}_{ij}^{(q)}$ 的函数。因此,要寻优的参数 $\bar{\omega}_{ij}^{(q)}$ 个数比较多,即目标函数 E 是关于连接

权的一个非常复杂的超曲面,这就给寻优计算带来一些问题。一个最大的问题是收敛速度慢;由于待寻优的参数太多,必然导致收敛速度慢的缺点。第二个严重缺陷是局部极值问题,即 E 的超曲面可能存在多个极值点。按照上述的寻优算法,它一般收敛到初值附近的局部极值,所以连接权系数的初值选取很重要。

8.2.3 BP 神经网络设计

从理论上说,只要有足够多的隐层和隐结点,即可实现复杂的映射关系。其中隐层结点数的选择是一个十分复杂的问题,目前还没有很好的解析式来表示,隐层结点数与问题的要求、输入输出单元的多少都有直接的关系。如何选取最佳的隐层结点数目,主要参考以下公式:

$$k < \sum_{i=0}^{n} C\binom{n_1}{i},\ n_1 = \sqrt{n+m} + a,\ n_1 = \operatorname{lb} n \tag{8-20}$$

其中,k 为样本数,n 为输入结点数,m 为输出样本数,n_1 为隐结点数。

神经网络的拓扑结构在一定程度上影响网络的分类能力。BP 网络能够实现输入输出的非线性映射关系,但它并不依赖于模型。其输入输出之间的关联信息分布地存储于连接权中。由于连接权的个数很多,个别神经元的损坏只对输入输出关系有较小的影响,因此 BP 网络具有较好的容错性。

BP 网络具有很好的逼近非线性映射的能力,因而它可应用于数据挖掘、信息处理、图像识别等多个方面。

8.3 自组织特征映射(SOFM)神经网络

1981 年芬兰学者 T. Kohonen 提出了自组织特征映射网络模型(self-organizing feature map,SOFM)网络模型。这种网络由可以自我调整连接强度的神经元组成,神经元的自调整过程和人脑的自组织过程相仿:各个神经元之间通过相互的侧向交互作用进行竞争,以近邻者相互激励、远邻者相互抑制的规则自适应地组织形成针对某类特殊信息的一种结构。自组织就是通过调整权重使神经网络收敛于一种表示形态,在这一表示形态中的一个神经元值对某种输入模式特别匹配或特别敏感。自组织特征映射的目的,就是使神经元的权重形态表示可以间接模仿输入的信号模式。

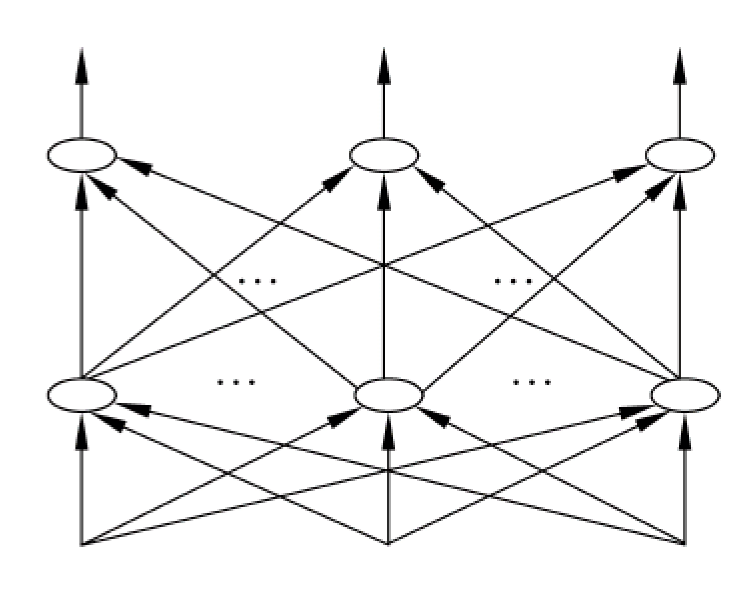
SOFM 神经网络对一个输入模式实施自学习算法,经过充分的学习后,网络各神经元的权重趋向于按照输入空间中样本的密度分布于该空间中,每一个神经元只对其权重周围一个小空间内的样本响应,形成输入空间到神经元集的一个映射,而且神经元之间形成一种特定的位置关系,使得在输入空间中相近的样本映射到输出层神经元平面相邻的神经元上。这样,SOFM 神经网络便通过自学习形成了一个对输入空间的内部表示。这种表示一方面反映原空间样本的密度分布,另一方面保持原空间样本之间的拓扑关系,这些信息被高度压缩到一个简单的有限个结点的平面上,使某些样本在空间中相对集中地聚集在一起,形成了聚类。

8.3.1 SOFM 神经网络的拓扑结构

SOFM 神经网络仅由输入层和输出层两层构成。输入层各神经元通过权向量将外界

信息汇集到输出层的各神经元。输入层的形式与 BP 网络相同，结点数目与样本维数相等。输出层也是竞争层，神经元的排列有多种形式，如一维线阵、二维平面和三维栅格阵。常见的是前两种。

如图 8-5 所示，网络上层是竞争层也是输出层，该层所有神经元按照任意一种非最优匹配形式排成一个邻域结构。根据该结构可以判别各神经元的领域内和该神经元的领域内有哪些神经元。网络下层为输入层，用于接收输入模式。


图 8-5　SOFM 神经网络的拓扑结构

8.3.2　SOFM 神经网络聚类的基本算法

SOFM 网络聚类的基础是自组织特征映射学习，运用侧向交互作用原理，在每个最优匹配神经元附近形成一个“聚类区”。学习的结果总是使聚类区内各个神经元的权重向量朝输入向量值逼近，从而使具有相近特性的输入向量聚集在一起。学习后的 SOFM 神经网络的权重能够代表输入向量的特征，把相似的输入向量归为同一类，并由权重的输出值来指示所代表的类别。SOFM 神经网络学习算法的核心是最优匹配神经元的选择和权重自组织过程。选择最优匹配神经元的实质是选择输入模式对应的中心神经元；权重的自组织过程则是根据中心神经元调整其邻域内神经元的权重，以“墨西哥帽”的形态来使输入模式得以存放。每进行一次学习，SOFM 神经网络就对外部输入模式执行一次自组织适应过程，其结果是强化现行模式的映射形态，弱化以往模式的映射形态。Kohonen 证明在学习结束时，每个权重向量都近似落入到由神经元所应类别的输入模式空间的中心。所以，向量可作为这个输入模式的最优参考向量。针对有 m 个样本的训练集，基本 SOFM 神经网络算法的步骤描述如下。

(1) 初始化。设计合理的网络结构，并将各个权重 $w_{ij}(0)$ 初始化为区间 $(0,1)$ 中的随机数，并设定最大迭代次数 $T(T>m)$ 和学习率及优胜邻域的初始值。

(2) 从训练集中选取一个输入模式 $X_k=[X_{1k},X_{2k},\cdots,X_{nk}]^{\mathrm{T}}$，其中 $k=1,2,\cdots$。

(3) 计算神经元 j 和 X_k 之间的距离 d_{jk}：

$$d_{jk}=\|X_k-w_j\|=\sqrt{\sum_{i=1}^{n}(X_{ik}-w_{ij})^2}\quad i=1,2,\cdots,n;j=1,2,\cdots,p \tag{8-21}$$

(4) 确定最优匹配神经元 C_t。C_t 是与 X_k 的距离最小的神经元，其权重 w_{ct} 满足：

$$\|X_k-w_{ct}\|=\min\{d_{jk}\} \tag{8-22}$$

(5) 计算最优匹配神经元 C_t 和优胜邻域 $N_c(t)$。优胜邻域的大小一半用邻域半径表示 $r(t)$。

$$r(t)=C_1\left(1-\frac{t}{T}\right) \tag{8-23}$$

其中，C_1 为与输出层结点数相关的正常数。

(6) 判断各个神经元是否在最优匹配神经元的领域内。

(7) 调整权重。按下式对 $N_c(t)$ 内的神经元进行权重调整：

$$w_{ij}=w_{ij}+\alpha(t)[X_i-w_{ij}] \tag{8-24}$$

$$\alpha(t) = C_2\left(1 - \frac{t}{T}\right) \tag{8-25}$$

其中，$\alpha(t)$表示学习率。且$\alpha(t)>0$对于$N_c(t)$外的神经元，其权重不变。C_2为0～1之间的常数。

(8) 当$t=T$或学习率和邻域减小到零值时，迭代结束并输出权重向量$w_j(t)$作为结果；否则，转步骤(2)。

8.3.3 SOFM 神经网络学习算法分析

在SOFM算法中，学习率和最优匹配神经元的领域$N_c(t)$都随迭代次数t的变化而发生变化，其初值设置和变化规律的选择是影响算法性能的关键所在。

1. 学习率

学习率$\alpha(t)>0$，在实际的使用中，最简单的方法是不设置初值$\alpha(0)$，直接经验时间递减函数。$\alpha(t)=1/t$作为迭代t次时的学习率。另外一种常用的方法是将初始的学习率$\alpha(0)$置一个较大的正小数，迭代t次时取$\alpha(t)=\alpha(0)/t$，但是，使用$\alpha(t)=1/t$或以$\alpha(t)=\alpha(0)/t$将导致随迭代次数t的增加学习率下降的速度比较陡峭的情况发生，如果将总迭代次数T和对学习过程的控制有机联系起来，结合当前迭代次数t与总迭代次数T来调整学习率，将能使学习率的下降速度比较平缓，表达为式(8-25)。学习率从开始值$\alpha(0)$按迭代次数的增加而逐渐减小，到迭代终止到达零值。

2. 邻域

邻域N_c是以神经元C为中心的一个方形或圆形区域，领域N_c的范围可由该区域所包含的神经元数目来确定。SOFM神经网络训练确定最优匹配神经元之后，需要先判断其他神经元是否落在最优匹配神经元邻域N_c内。最后，再对落在最优匹配神经元领域N_c内的神经元进行权重调整。邻域N_c是时变的，初始阈值$N_c(0)$一般包含整个输出神经元阵列。随迭代次数t的增加，领域N_c向以最优匹配神经元为中心的小范围单调变小，最后应缩小到只包含单独的一个最优匹配神经元。随迭代次数的增加，所考虑的领域N_c变小；表达为式(8-23)。

8.4 Elman 神经网络

Elman神经网络是Elman于1990年提出的，该模型在前馈网络的隐含层中增加一个承接层，作为一步延时算子，达到记忆的目的，从而使系统具有适应时变特性的能力，能直接反映动态过程系统的特性。

8.4.1 Elman 神经网络的拓扑结构

Elman神经元网络一般分为4层：输入层，中间层(隐含层)，承接层和输出层，如图8-6所示。其输入层，隐含层和输出层的连接类似于前馈网络，输入层的单元仅起信号输入作用，输出层单元起线性加权作用。隐含层单元的传递函数可采用线性或非线性函数，承接层又称为上、下层或状态层，它用来记忆隐含层单元前一时刻的输出值，可以认为是一个一步延时算子。

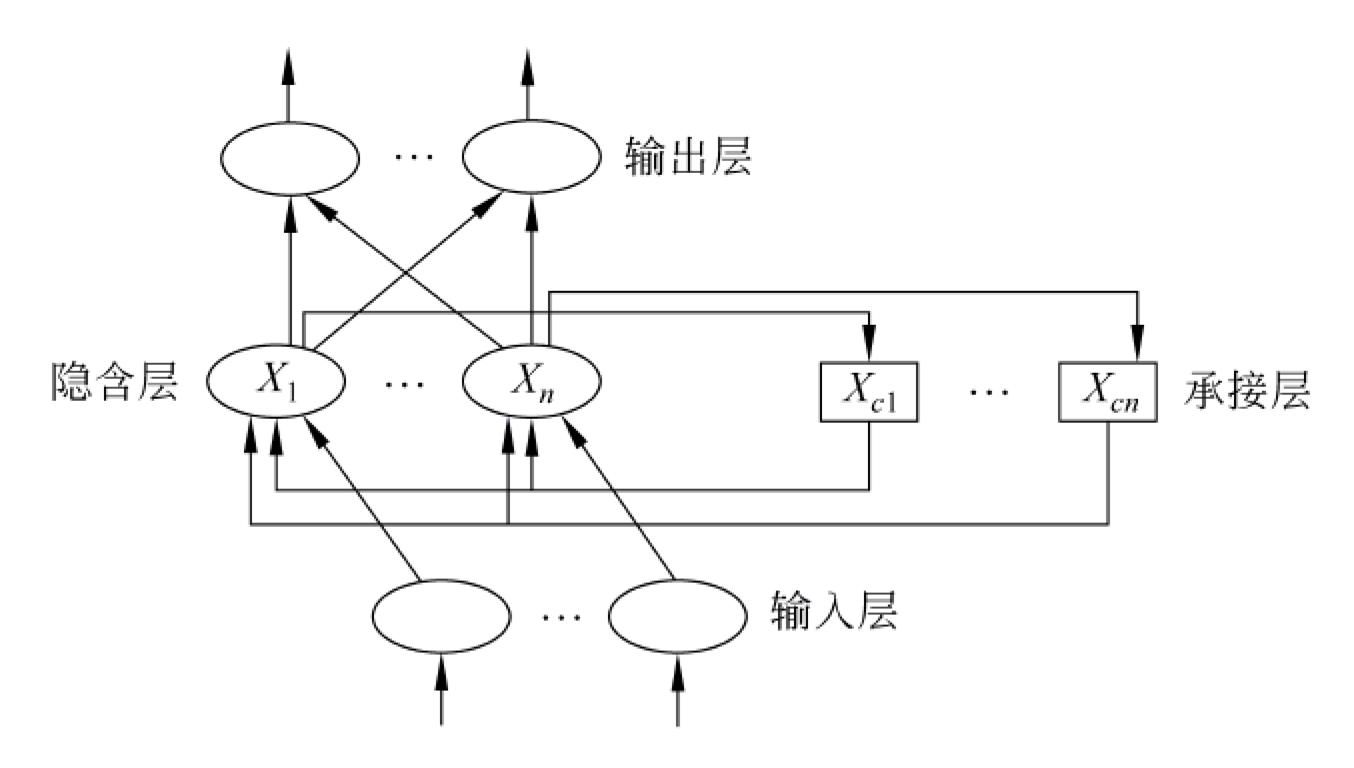


图 8-6　Elman 神经网络的拓扑结构

Elman 神经元网络的特点是隐含层的输出通过承接层的延迟与存储，自联到隐含层的输入，这种自联方式使其对历史状态的数据具有敏感性，内部反馈网络的加入增加了网络本身处理动态信息的能力，从而达到动态建模的目的。

8.4.2　Elman 神经网络权值计算

Elman 网络的非线性状态空间表达式为

$$Y(k) = g(W^3 X(k)), \quad X(k) = f(W^1 X_c(k) + W^2 U(k-1)),$$
$$X_c(k) = X(k-1) \tag{8-26}$$

其中，Y,X,U,X_c 分别表示 m 维输出点向量，n 维中间层结点单元向量，r 维输入向量和 n 维反馈状态向量。W^3,W^2,W^1 分别表示隐含层到输出层、输入层到隐含层、承接层到隐含层的连接权值，$g(\cdot)$为输出神经单元的传递函数，是隐含层输出的线性组合，$f(\cdot)$为中间层神经元的传递函数，常采用 S 型函数。

Elman 网络也采用 BP 算法进行权值修正学习指标函数误差平方和函数：

$$E(W) = \sum [Y_k(W) - \bar{Y}_k(W)]^2, \quad k = 1,2,\cdots,n \tag{8-27}$$

其中，$\bar{Y}_k(W)$为目标输出向量。

具体计算方法参见第 8.2.2 小节的 BP 算法。另隐含层结点个数的确定也可参考第 8.2.3小节的公式(8-20)。

8.5　Hopfield 神经网络

Hopfield 神经网络是 1982 年由美国加州理工学院物理学家 J. J. Hopfield 教授提出的，在神经网络中引入了“能量函数”的概念，是最著名且应用广泛的反馈神经网络。反馈神经网络是指拓扑结构中有环路的神经网络，如图 8-3(b)所示。反馈环路的存在，使得网络的输出部分影响作用于网络的输入，使网络产生动态特性，对网络的学习能力和性能产生深刻的影响。所有结点都是计算结点，同时也可接受输入，并向外界输出。即在反馈网络中，信息在向前传递的同时还要反向反馈，这种信息反馈可以发生在不同网络层神经元之间，也可以只局限于某一层神经元上。

反馈型神经网络如果总结点(神经元)数为 N，那么每个结点有 N 个输入和 1 个输出，

也就是说，所有结点都是一样的，它们之间都可相互连接。它是一种反馈动力学系统，它需要工作一段时间才能达到稳定。在反馈型神经网络结构中，整个神经网络的输入的值是由其输出 $y_i, i \in [1, N]$ 通过某种反馈机制计算得到的。

8.5.1 Hopfield 神经网络的拓扑结构

Hopfield 神经网络是全连接的反馈网络，其网络结构如图 8-7 所示，即网络每一次的演化结果都重新作为网络的输入，并且重新演化，主要用于联想记忆和优化计算等。

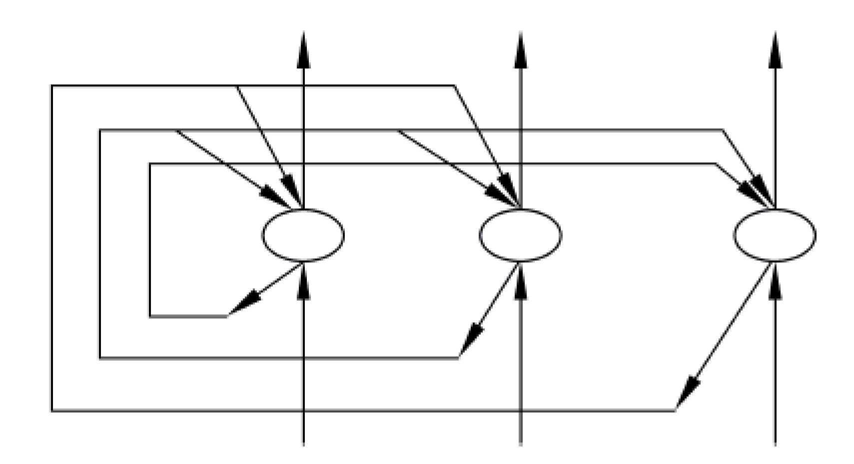


图 8-7　Hopfield 神经网络的拓扑结构

联想记忆是指当网络输入某个矢量后，网络经过反馈演化，从网络输出端得到另一个矢量，这样的输出矢量称作网络从初始输入矢量联想得到的一个稳定记忆，即网络的一个平衡点。优化计算是指当某一问题存在多种算法时，可以设计一个目标函数，然后寻求满足这一目标函数的最优解法。例如，在很多情况下可以把能量函数作为目标函数，得到的最优解法需要能使能量函数达到极小点，即能量函数的稳定平衡点。总之，Hopfield 网络的设计思想就是在初始输入下，使网络经过反馈计算最后达到稳定状态，这时的输出就是用户需要的平衡点。

8.5.2 Hopfield 神经网络学习算法概述

设用 $X(t)$ 表示网络在时刻 t 的状态，如果从 $t=0$ 的任一初态 $X(0)$ 开始，存在一个有限的时刻 t，使得从此时刻开始，神经网络的状态不再发生变化，即

$$X(t+\Delta t) = X(t), \quad \Delta t > 0 \tag{8-28}$$

就称此网络是稳定的。Hopfield 神经网络主要有以下几个特点。

(1) 具有对称的神经元连接权值矩阵 W，即 $W_{ij} = W_{ji}$。

(2) 无自反馈，即 $W_{ii} = 0$。

(3) 系统有稳定解，不同的状态有可能收敛于稳定状态，又称不动点吸引子；或者发生震荡，即收敛于周期二吸引子。Marcus 等人证明具有对称权值矩阵的系统只有不动点周期二吸引子和吸引子的解。

8.5.3 离散 Hopfield 神经网络

离散神经网络模型是一个离散时间系统，每个神经元只有两个状态，可以用 1 和 0 来表示，由连接权重 W_{ij} 所构成的矩阵是一个对角线为 0 的对称矩阵，即

$$W_{ij} = \begin{cases} W_{ji}, & i \neq j \\ 0, & i = j \end{cases} \tag{8-29}$$

如果用 $x(t)$ 表示整个网络在时刻 t 的状态，则 X 代表网络中每个人工神经元的状态。所以，状态向量 X 中的分量个数就是网络中人工神经元的个数。假设网络中的结点个数为 n，则向量 X 的构成如下：

$$X^T(t) = [x_1(t), x_2(t), \cdots, x_i(t), \cdots, x_n(t)] \tag{8-30}$$

$x_i(t)$ 表示结点 i 在时刻 t 的状态，该结点在时刻 $t+1$ 的状态由下式决定：

$$X_i(t+1)=\begin{cases}1, & H_i(t)\geqslant 0\\ 0, & H_i(t)<0\end{cases} \tag{8-31}$$

这里，$H_i(t)=\sum_{i=1}^{n}W_{ij}X_i(t)-\theta_j$，其中的 W_{ij} 为 i 到结点 j 的连接权重；θ_j 为结点 j 的阈值。

离散型 Hopfield 网络有两种工作模式。

(1) 串行方式，是指在任一时刻 t，只有一个神经元 i 发生状态变化，而其余的神经元保持状态不变。

(2) 并行方式，是指在任意时刻 t，都有部分或全体神经元同时改变状态。

有关离散 Hopfield 网络的稳定性问题，已于 1983 年由 Cohen 和 Gross berg 给予了证明。而 Hopfield 等人又进一步证明，只要连接权值构成的矩阵是非负对角的对称矩阵，则该网络就具有串行稳定性。

8.5.4 连续 Hopfield 神经网络

Hopfield 网络是一种非线性的动力网络，可通过反复的网络动态迭代来解决问题。在求解某些问题时，其求解问题的方法与人类求解问题的方法很相似，虽然所求得的解不一定是最佳解，但其求解速度快，更符合人们日常解决问题的策略。

1984 年，Hopfield 提出了连续时间的神经网络，在这种神经网络中，各结点可在 0 和 1 之间内取任一实数值。连续型 Hopfield 网络的输入和输出为连续可微且单调上升的函数，每个神经元的输入是一个随时间变化的状态变量，与外界输入和从其他神经元来的偏置信号有直接关系，同时也与其他神经元与自身之间的连接权有关系。状态变量直接影响输入变量，使系统变成一个随时间变化的动态系统。

连续型 Hopfield 神经网络在网络的结构上与离散型相同，且状态方程形式上也相同。连续型 Hopfield 神经网络状态的演变过程用微分方程描述如下。

1. 设置互连权值

$$W_{ij}=\begin{cases}\sum_{s=0}^{m-1}x_i^s x_j^s, & i\neq j\\ 0, & i=j\end{cases} \tag{8-32}$$

其中，x_i^s 是 s 类样例的第 i 个分量，它可以为 1 和 0，样例类别数为 m，结点数为 n。

2. 未知类别初始化

$$y_i(0)=x_i,\quad 0\leqslant i\leqslant n-1 \tag{8-33}$$

其中，$y_i(t)$为结点 i 在 t 时刻的输出，当 $t=0$ 时，$y_i(0)$就是结点 i 的初始值，x_i 为输入样本的第 i 个分量。

3. 迭代直到收敛

$$y_j(t+1)=f\Big(\sum_{i=0}^{n-1}W_{ij}y_i(t)\Big),\quad 0\leqslant j\leqslant n-1 \tag{8-34}$$

其中 f 为阈值型激发函数。该过程一直迭代到不再改变结点的输出为止。这时，各结点的输出与输入样例达到最佳匹配。否则转 2 继续。

前面已经指出，当 Hopfield 模型的网络中各神经元的连接权值所构成的矩阵是一个非负对角的对称矩阵，或者是一个非负定矩阵时，上述算法都是收敛的。

8.6 利用 SQL Server 2005 神经网络进行数据挖掘

Microsoft 神经网络支持 Microsoft 决策树可以执行的所有任务，包括分类、回归和关联。前两个任务是神经网络最常见的任务，而关联任务可能太耗时和耗资源，所以一般不推荐使用神经网络。

以下是 Microsoft 神经网络算法的几个参数。

（1）MAXIMUM_INPUT_ATTRIBUTES 参数。用于特征选择的阈值参数。当输入属性的数目大于该参数的设置时，会隐式的调用特征选择来选择最重要的属性。

（2）MAXMUIM_OUTPUT_ATTRIBUTES 参数。用于特征选择的阈值参数。当可预测的属性数目大于该参数的设置时，会隐式的调用特征选择来选择最重要的属性。

（3）MAXIMUM_States 参数。它指定算法所支持的属性状态数目的最大值。如果一个属性所拥有的状态数目大于状态数目的最大值，则算法会使用属性的出现最频繁的状态，然后将剩下的状态认为是缺失状态。

（4）Holdout_Percentage 参数。它指定测试数据的百分比。测试数据用于在训练期间验证正确性。默认值为 0.1。

（5）Holdout_Seed 参数。它是一个整数，用来指定种子，该种子用于选择测试数据集。

（6）Hidden_Node_Ratio 参数。它用于配置隐含结点的数目。隐含结点的基数为 sqrt($(n\times m)$)，其中 n 是输入神经元的数目，m 是输出神经元的数目。如果 Hidden_Node_Ratio 等于 2，则隐含结点数目等于 $2\times$sqrt$(n\times m)$。Hidden_Node_Ratio 的默认值是 4。

（7）Sample_Size 参数。它指定用于训练的事例数目的上限，默认值为 10000。

（8）Microsoft 逻辑回归算法参数。它是基于 Microsoft 神经网络算法的实现，也就是将 Microsoft 神经网络算法的参数 Hidden_Node_Ratio 设置为 0，Microsoft 神经网络算法就变成了 Microsoft 逻辑回归算法。如果使用 Microsoft 神经网络来构建一个没有隐含层的模型，则会获得与使用 Microsoft 逻辑回归相同的结果。

8.6.1 数据准备

在进行数据挖掘之前，需要建立“数据源”和“数据源视图”。本章的实例依然沿用第 5 章和第 6 章中的“商业银行信贷”数据库，对数据库的处理也同第 6.5.1 小节的内容，此外，还做了如下处理。

对该数据库进行了以下数据类型转换处理，将 t_dm 表中用作输入列(可参见表 6-4)的文本类型数据转换成数值类型，来满足就神经网络对数据的要求。下面以输入列“经济性质”为例，来说明具体转换过程，其他例的转换也是类似的。

（1）统计输入列中不同类别的个数，SQL 实现语句及执行结果如图 8-8 所示。

（2）根据第(1)步统计信息用数值来替代文本数据，SQL 实现语句及执行结果如图 8-9 所示。

```
select count(*),经济性质from t_dm group by 经济性质
```

(a) 统计类别SQL语句

	(无列名)	经济性质
1	187	个体
2	1	研究所
3	408	股份合作
4	36	民营
5	208	其他
6	72	三资
7	1	部队
8	18	学校
9	1262	国有
10	419	集体
11	1	联营
12	55	其他股份制
13	13	外贸
14	15	医院
15	38	机关团体
16	496	国有控股
17	106	私营
18	74	集体控股

(b) 执行结果

图 8-8　统计类别个数的实现

```
update t_dm set 经济性质='1' where 经济性质='国有'
update t_dm set 经济性质='2' where 经济性质='其他股份制'
update t_dm set 经济性质='3' where 经济性质='其他'
update t_dm set 经济性质='4' where 经济性质='国有控股'
update t_dm set 经济性质='5' where 经济性质='民营'
update t_dm set 经济性质='6' where 经济性质='集体'
update t_dm set 经济性质='7' where 经济性质='外贸'
update t_dm set 经济性质='8' where 经济性质='私营'
update t_dm set 经济性质='9' where 经济性质='三资'
update t_dm set 经济性质='10' where 经济性质='股份合作'
update t_dm set 经济性质='11' where 经济性质='集体控股'
update t_dm set 经济性质='12' where 经济性质='研究所'
update t_dm set 经济性质='13' where 经济性质='机关团体'
update t_dm set 经济性质='14' where 经济性质='医院'
update t_dm set 经济性质='15' where 经济性质='学校'
update t_dm set 经济性质='16' where 经济性质='个体'
update t_dm set 经济性质='17' where 经济性质='部队'
update t_dm set 经济性质='18' where 经济性质='联营'
```

(a) 更新表中数据SQL语句

```
消息
(1262 行受影响)
(55 行受影响)
(208 行受影响)
(496 行受影响)
(36 行受影响)
(419 行受影响)
(13 行受影响)
(106 行受影响)
(72 行受影响)
(408 行受影响)
(74 行受影响)
(1 行受影响)
(38 行受影响)
(15 行受影响)
(18 行受影响)
(187 行受影响)
(1 行受影响)
```

(b) 执行结果

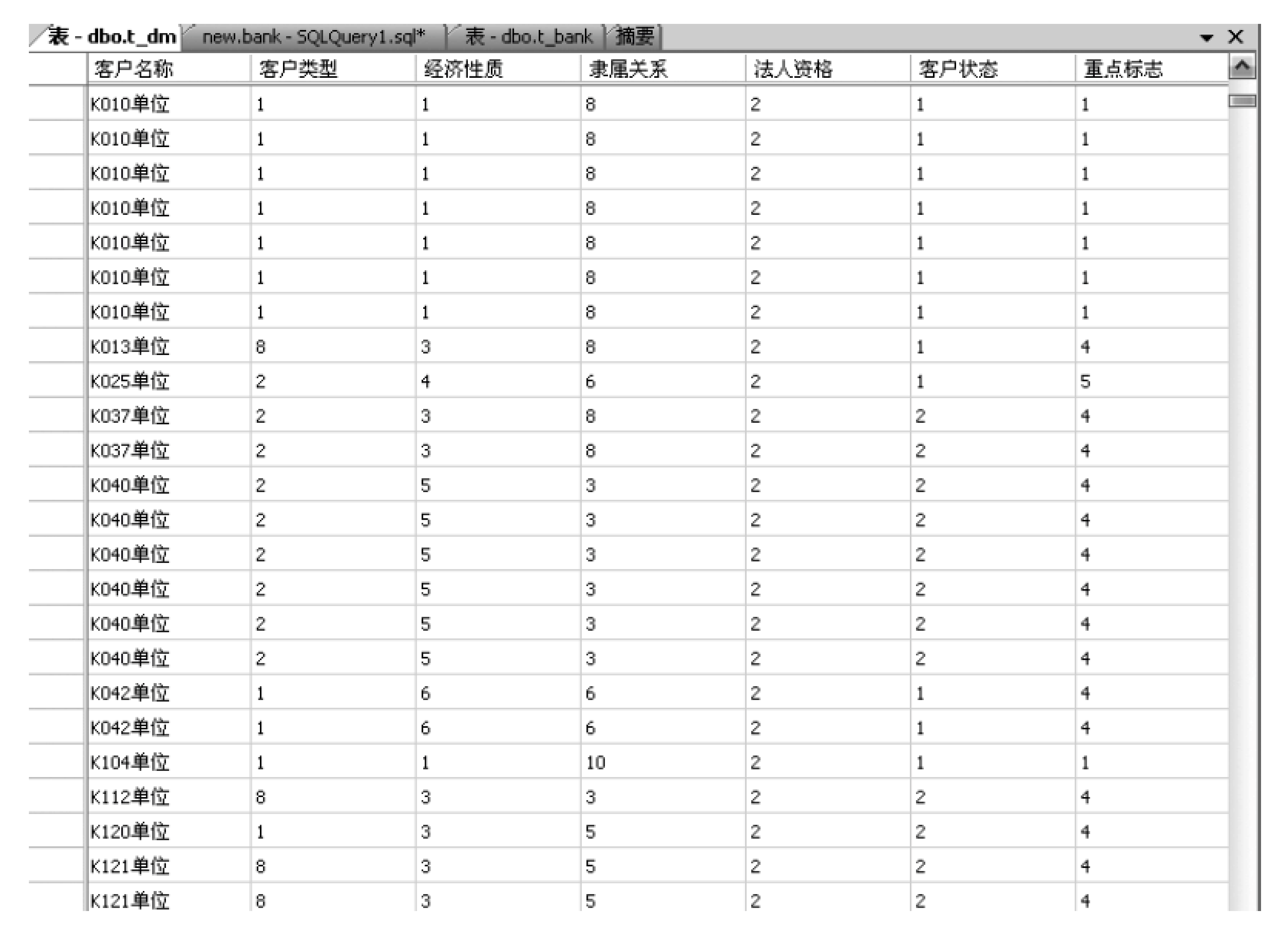
图 8-9　更新表中数据

所有输入列经过以上两步处理之后，部分 t_dm 表中数据如图 8-10 所示。

8.6.2　挖掘流程

(1) 右击项目 Neutral Network 下的“挖掘结构”，从弹出的快捷菜单中选择“新建挖掘结构”，打开“数据挖掘向导”对话框，单击“下一步”按钮，弹出“选择定义方法”对话框，单击“下一步”按钮，弹出“选择数据挖掘技术”对话框。

(2) 在如图 8-11 所示的对话框中，下拉列表框中选取“Microsoft 神经网络”选项，单击“下一步”按钮，进行下一步操作。



客户名称	客户类型	经济性质	隶属关系	法人资格	客户状态	重点标志
K010单位	1	1	8	2	1	1
K010单位	1	1	8	2	1	1
K010单位	1	1	8	2	1	1
K010单位	1	1	8	2	1	1
K010单位	1	1	8	2	1	1
K010单位	1	1	8	2	1	1
K010单位	1	1	8	2	1	1
K013单位	8	3	8	2	1	4
K025单位	2	4	6	2	1	5
K037单位	2	3	8	2	2	4
K037单位	2	3	8	2	2	4
K040单位	2	5	3	2	2	4
K040单位	2	5	3	2	2	4
K040单位	2	5	3	2	2	4
K040单位	2	5	3	2	2	4
K040单位	2	5	3	2	2	4
K040单位	2	5	3	2	2	4
K042单位	1	6	6	2	1	4
K042单位	1	6	6	2	1	4
K104单位	1	1	10	2	1	1
K112单位	8	3	3	2	2	4
K120单位	1	3	5	2	2	4
K121单位	8	3	5	2	2	4
K121单位	8	3	5	2	2	4

图 8-10　经处理的 t_dm 数据示意图



图 8-11　选择数据挖掘技术

(3) 如图 8-12 所示，在“选择数据源视图”页面的“可用数据源视图”列表中显示了前面步骤创建的 bank 数据源视图，选中该视图选项，单击“下一步”按钮，进行下一步操作。



图 8-12　选择数据源视图

(4) 如图 8-13 所示，在“指定表类型”页面中可以看到 bank 数据源视图包含的数据表，勾选“t_dm”选项右边的“事例”复选框，可以将其定义为事例表；单击“下一步”按钮，进行下一步操作。



图 8-13　指定表类型

(5) 如图 8-14 所示，在“指定数据类型”页面显示了挖掘模型结构，在各个选项右边勾选不同的复选框，可参照表 6-4 完成，然后单击“下一步”按钮，进行下一步操作。

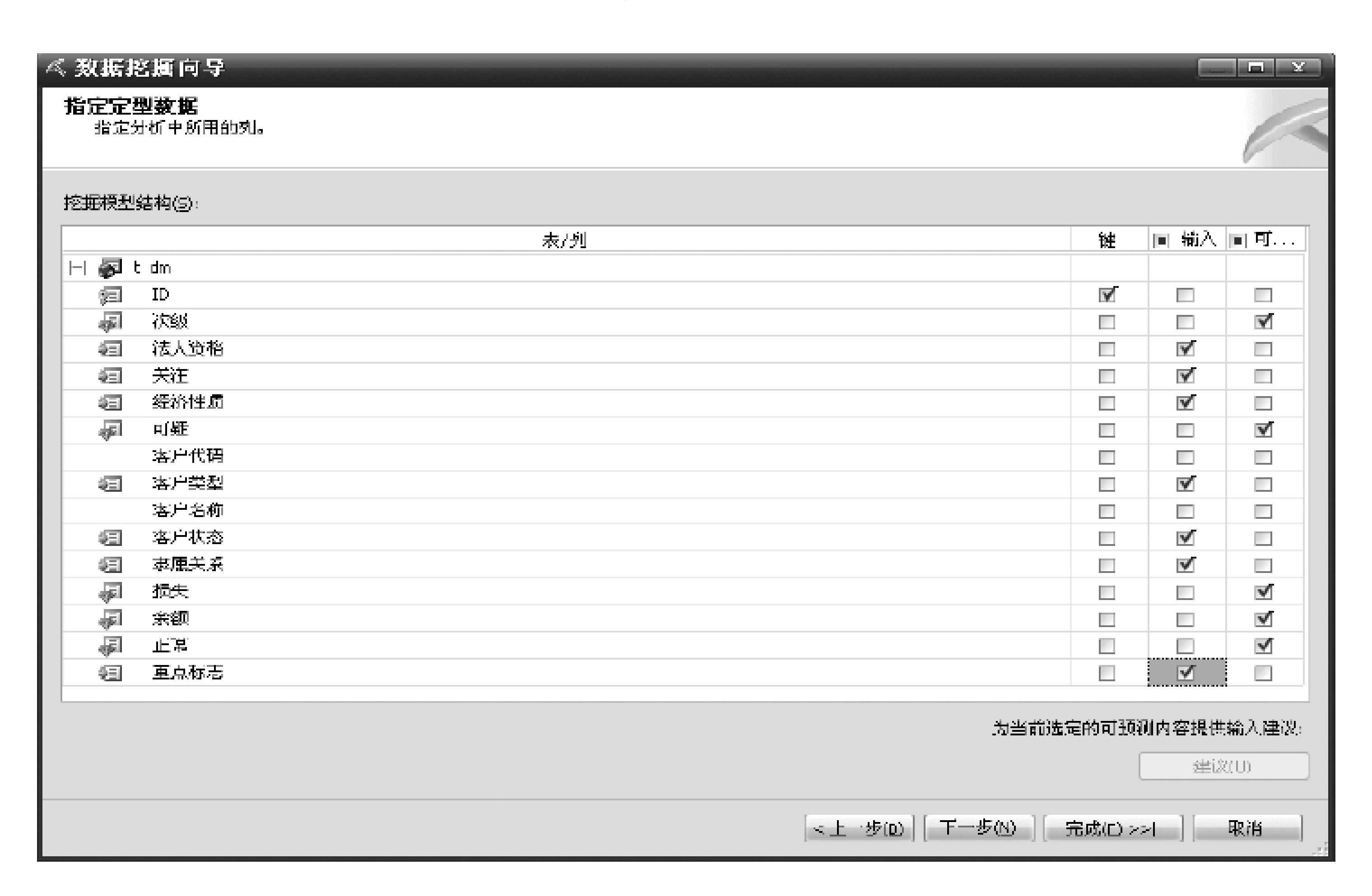


图 8-14　指定数据类型

(6) 如图 8-15 所示，经过“检测”将指定数字列，即“次级”、“关注”、“可疑”、“损失”、“余额”和“正常”的连续值转换成离散值，即 0 或 1，也与第 6.5.1 小节中做的数据处理对应起来。在“指定列的内容和数据类型”页面中显示了指定“ID”的内容类型为 Key，“余额”的内容类型为 Continuous，其余列内容类型均为 Discrete；ID 的数据类型为 long，其余各列数据类型均为 Double，单击“下一步”按钮，进行下一步操作。

(7) 如图 8-16 所示，在“完成向导”页面中将数据挖掘结构命名为 t_Dm1，单击“完成”按钮，完成挖掘结构的创建。

(8) 单击“挖掘准确性图表”选项卡下的“提升图”和“分类矩阵”，其结果如图 8-17 和图 8-18 所示。

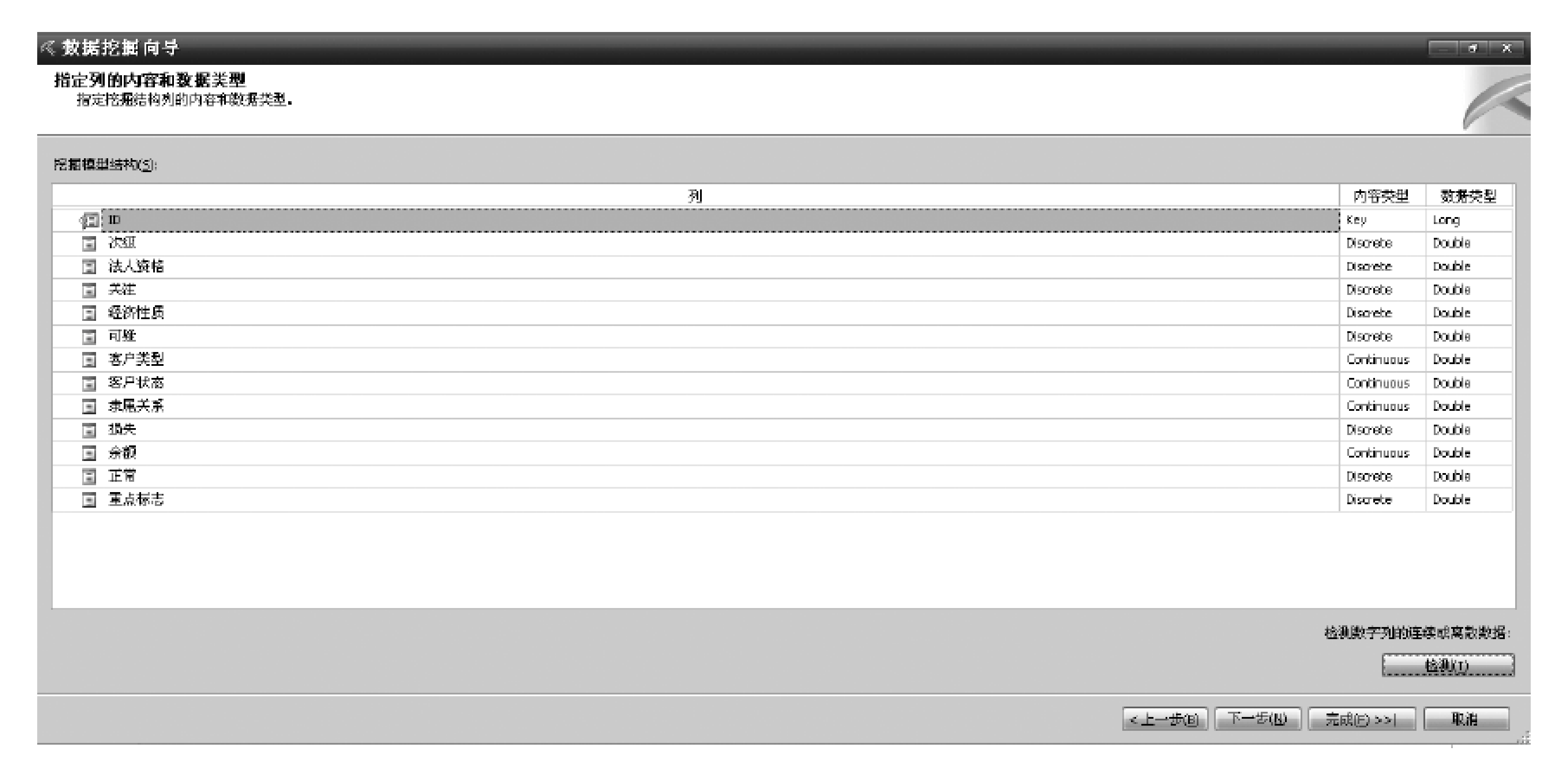


图 8-15　指定列的内容和数据类型



图 8-16　完成数据挖掘结构的创建

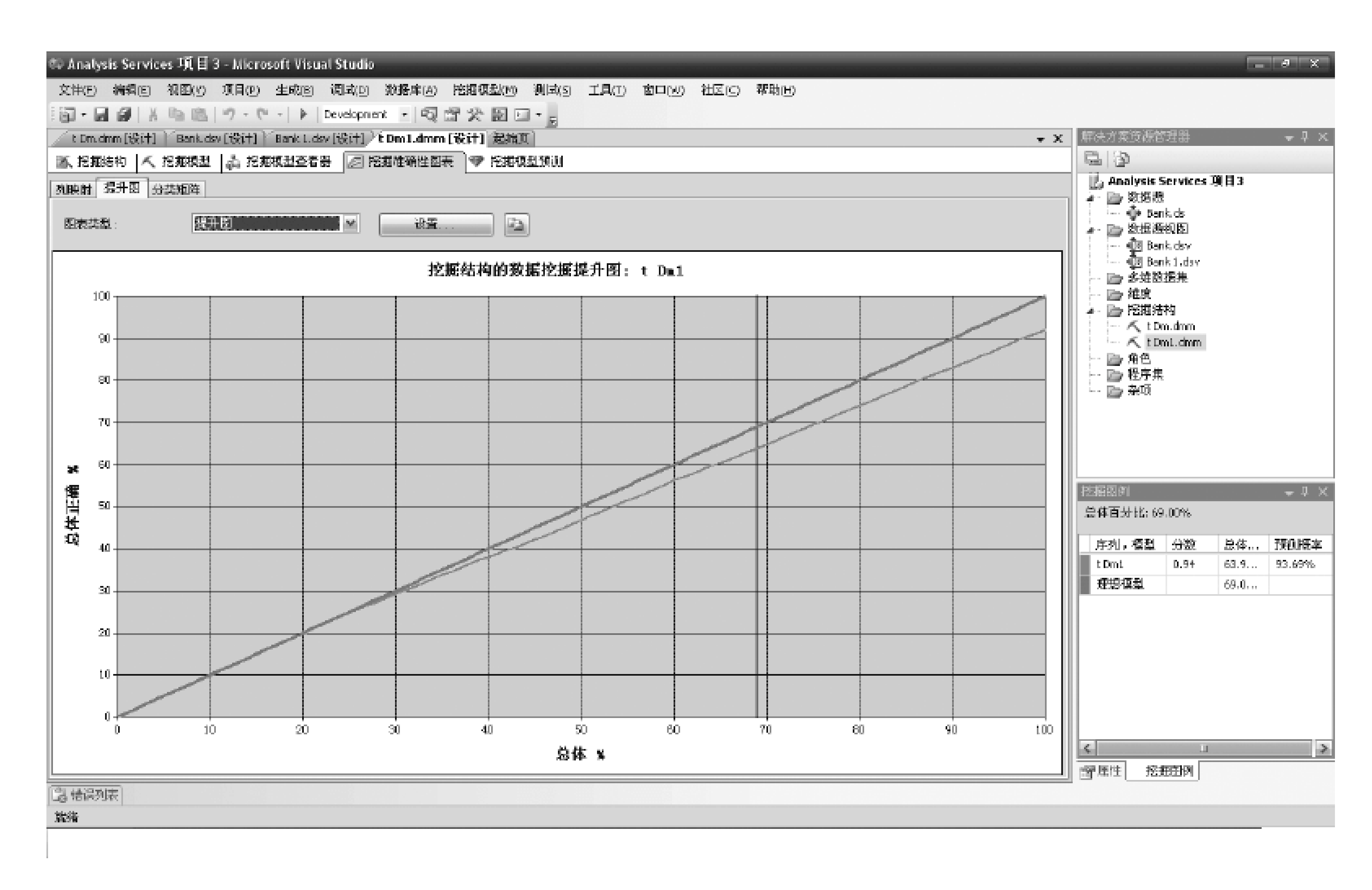


图 8-17　“次级”提升图

图 8-18 中上方的线代表神经网络建立的预测模型，下方的线代表实际模型，可以看出两者是基本匹配的，说明预测模型比较理想；图 8-18 中，对“次级”=1 的预测结果是无差错的，对“次级”=0 的预测结果有 7.9%的误差。

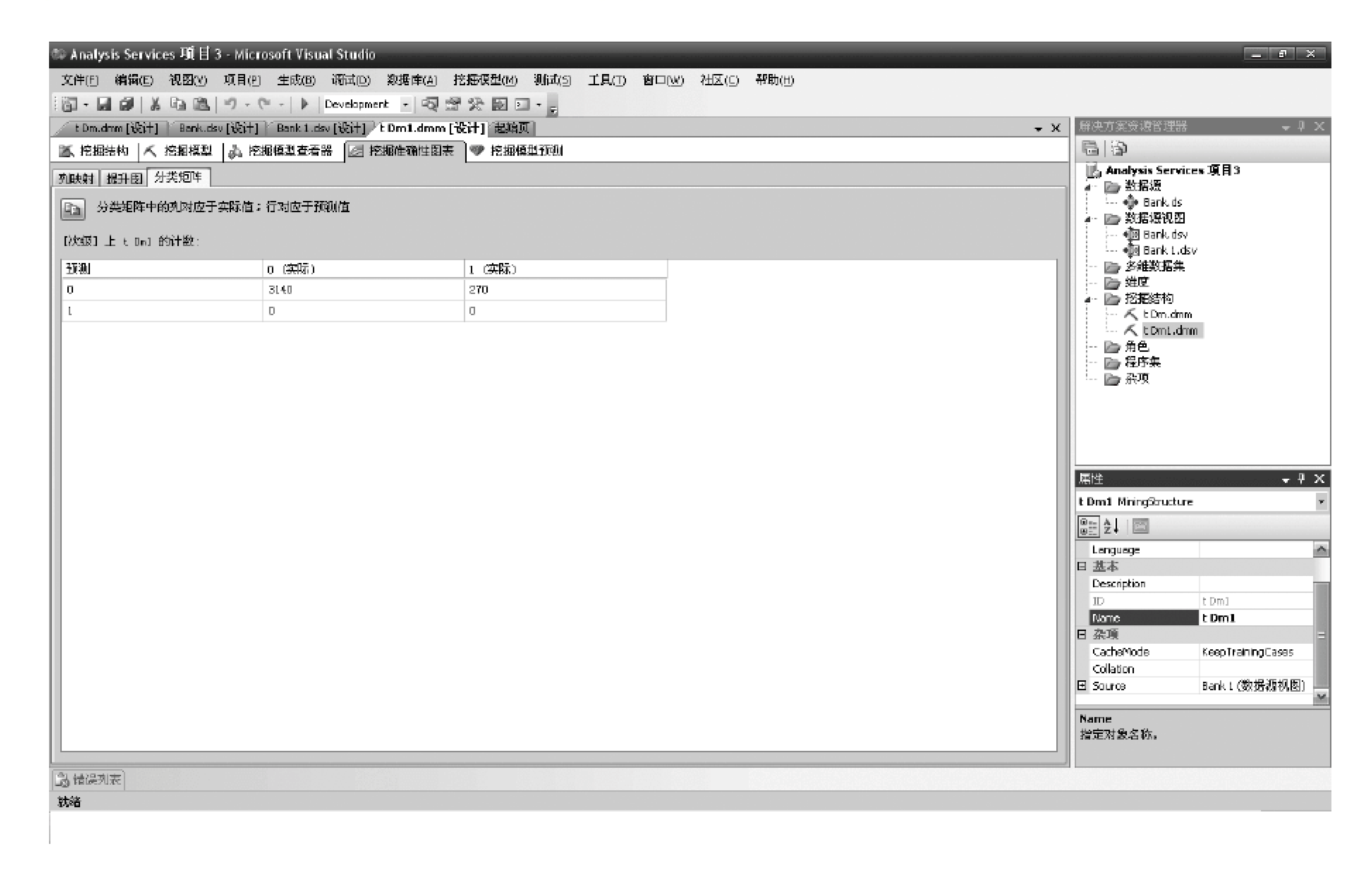


图 8-18 “次级”分类矩阵图

小结

本章介绍了数据挖掘的智能方法-人工神经网络。首先介绍了人工神经网络的基本原理，包括人工神经元的基本构成和各种的学习规则；然后分别介绍了数据挖掘中常用的 BP、神经网络、SOFM 神经网络、Elman 神经网络和 Hopfield 神经网络 4 种网络，主要介绍它们的神经元特性、网络拓扑结构和学习算法三要素；最后通过实例说明了在 SQL Server 2005 中利用神经网络进行数据挖掘的过程。

习题 8

1. BP 神经网络有哪些优缺点？试各列举 3 条。
2. 人工神经元有哪些性质？
3. SOFM 神经网络的学习算法步骤？
4. BP 神经网络与 SOFM 神经网络的异同？
5. 人工神经网络常用的学习规则有哪些？

第 9 章　聚 类 分 析

聚类(Clustering)分析是数据挖掘技术的重要组成部分,它能从潜在的数据中发现有意义的数据分布模式,已经广泛应用于模式识别、数据分析、图像识别及其他许多方面。聚类是在事先不规定分组规则的情况下,将数据按照其自身特征划分成不同的群组。其重要特征是“物以类聚”,即要求在同一类的数据对象尽可能的相似,在不同类的数据对象尽可能的相异。聚类和分类的根本区别在于:分类需要事先知道所依据的对象特征,而聚类是在不知道对象特征的基础上要找到这个特征。聚类分析的方法很多,其中包括基于划分的聚类方法、基于层次的聚类方法、基于密度的聚类方法、基于网格的聚类方法和谱聚类方法等。

9.1　聚类概述

9.1.1　聚类简介

将物理或抽象对象的集合分组成为由类似的对象组成的多个类的过程被称为聚类。由聚类所生成的簇是一组数据对象的集合,这些对象与同一个簇中的对象彼此相似,与其他簇中的对象相异。在许多应用中,可以将一个簇中的数据对象作为一个整体来对待。

聚类分析已经广泛地应用于模式识别、数据分析、图像处理以及市场研究等许多领域中。在商务领域,聚类能帮助市场分析人员从客户基本库中发现不同的客户群,并且用购买模式来刻画不同客户群的特征。在生物学领域,聚类能对基因进行分类,获得对种群中固有结构的认识。聚类也能用于对 Web 文档进行分类,以发现信息。

作为一个数据挖掘的功能,聚类分析能作为一个独立的工具来获得数据分布的情况,观察每个簇的特点,集中对特定的某些簇做进一步分析。此外,聚类分析可以作为其他算法(如特征选择和分类等)的预处理步骤。

9.1.2　聚类的定义

在数据空间 $\mathscr{A}$ 中,数据集 $\boldsymbol{X}$ 由许多数据点(或数据对象)组成,数据点 $\boldsymbol{X}_i=(X_{i1},\cdots,X_{id})\in\mathscr{A}$,$\boldsymbol{X}_i$ 的每个属性 X_{ij} 既可以是数值型的,也可以是枚举型的。假设数据集 $\boldsymbol{X}$ 中有 N 个对象 $\boldsymbol{X}_i(i=1,2,\cdots,N)$,数据集 $\boldsymbol{X}$ 相当于一个 $N\times d$ 的矩阵。聚类的最终目的就是把数据集 $\boldsymbol{X}$ 划分为 k 个分割 $C_m(m=1,2,\cdots,k)$,也可能有些对象不属于任何一个分割,这些就是噪声 C_n。所有这些分割与噪声的并集就是数据集 $\boldsymbol{X}$,并且这些分割之间没有交集,即

$$\begin{cases}\boldsymbol{X}=C_1\cup\cdots\cup C_k\cup C_n\\ C_i\cap C_j=\varnothing\quad(1\leqslant i,j\leqslant k,i\neq j)\end{cases}\tag{9-1}$$

这些分割 $C_m(m=1,2,\cdots,k)$就是聚类。

9.1.3　聚类的要求

数据挖掘对聚类算法有如下要求。

1. 可伸缩性

许多聚类算法在小于200个数据对象的小数据集合上工作得很好。但是一个大规模数据库往往可能包含几百万个对象，一些聚类算法在这样的大数据集合上进行聚类分析可能会得到有偏差的结果。所以一个好的聚类算法应具有高度可伸缩性。

2. 处理不同类型属性的能力

聚类通常用于处理数值型数据，但也有可能被要求处理其他类型的数据，如二元类型，分类/标称类型，序数型数据，或者这些数据类型的混合。所以一个好的聚类算法应具有处理不同类型属性的能力。

3. 发现任意形状的聚类

许多聚类算法基于欧几里得距离或曼哈坦距离。基于距离度量的算法趋于发现具有相近尺度和密度的球状簇。但是，一个簇可能是任意形状的，因此提出能发现任意形状簇的聚类算法是很重要的。

4. 使输入参数的领域知识最小化

许多聚类算法在聚类分析中要求用户输入一些参数，例如希望产生的簇的数目。聚类结果对于输入参数十分敏感，然而参数通常很难确定，特别是对于包含高维对象的数据集来说，更是如此。一个好的聚类算法应包含尽量少的输入参数。

5. 处理噪声数据的能力

绝大多数的数据库都包含了孤立点、空缺、未知数据或错误的数据。一些聚类算法对这样的数据敏感，可能导致低质量的聚类结果。因此一个好的聚类算法应该具备较强地处理噪声的能力。

6. 对于输入记录的顺序不敏感

一些聚类算法对于输入数据的顺序敏感。例如，同一个数据集合，当以不同的顺序提交给同一个算法时，可能生成差别很大的聚类结果。因此对数据输入顺序的不敏感也是一个好的聚类算法所应该具备的。

9.2 聚类分析中的相异度计算

9.2.1 聚类算法中的数据结构

假设要聚类的数据集合包含 n 个数据对象，这些数据对象可能表示人、房子、文档、国家等。聚类算法通常采用如下两种具有代表性的数据结构。

1. 数据矩阵(或对象与变量结构)

数据矩阵是用 p 个变量(也称为度量或属性)来表现 n 个对象，例如用学号、姓名、年龄、性别、民族、所在系等属性来表现对象“学生”。这种数据结构是关系表的形式，或看作 $n\times p$(n 个对象$\times p$ 个变量)的矩阵。

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \tag{9-2}$$

2. 相异度矩阵(或对象-对象结构)

相异度矩阵存储 n 个对象两两之间的近似性,表现形式是一个 $n\times n$ 维的矩阵。

相异度矩阵元素 $d(i,j)$是对象 i 和对象 j 之间相异性的量化表示,通常它是一个非负的数值,当对象 i 和对象 j 越相似或接近时,其值越接近 0;两个对象越不同,其值越大。其中 $d(i,j)=d(j,i)$,$d(i,i)=0$,则可以得到如下的矩阵表示。

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & d(n,3) & \cdots & 0 \end{bmatrix} \tag{9-3}$$

数据矩阵因为其行和列代表不同的实体,经常被称为二模矩阵;而相异度矩阵因为其行和列代表相同的实体,被称为单模矩阵。许多聚类算法以相异度矩阵为基础,所以如果数据是用数据矩阵的形式表现的,在使用算法之前要将其转化为相异度矩阵。

下面将详细介绍相异度的计算问题。首先介绍如何计算用区间标度变量,二元变量,标称、序数和比例标度变量,或这些变量类型的组合来描述的对象相异度。

9.2.2 区间标度变量及其相异度计算

1. 区间标度变量

区间标度变量是一个粗略线性标度的连续变量。典型的例子包括重量和高度、经度和纬度坐标以及大气温度等。

选用度量单位将直接影响聚类分析的结果。例如,将高度的度量单位由“米”改为“英寸”,或者将重量的单位由“千克”改为“磅”,可能产生非常不同的聚类结构。一般而言,所用的度量单位越小,变量可能的值越大,这样对聚类结果的影响也越大。

为了避免对度量单位选择的依赖,数据应当标准化。标准化度量值试图给所有的变量相等的权重。当没有关于数据的先验知识时,这样做是非常有用的。

为了实现度量值的标准化,当给定一个变量 f 的值后,可以进行如下的变换。

(1) 计算平均的绝对偏差 S_f。

$$S_f = \frac{1}{n}(| x_{1f} - m_f |+| x_{2f} - m_f |+ \cdots +| x_{nf} - m_f |) \tag{9-4}$$

其中,$x_{1f}\cdots x_{nf}$ 是 f 的 n 个度量值,m_f 是 f 的平均值,即

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf}) \tag{9-5}$$

(2) 计算标准化的度量值 z-score。

$$z_{if} = \frac{x_{1f} - m_f}{s_f} \tag{9-6}$$

2. 相异度计算

在标准化处理后,或者在某些应用中不需要标准化,对象间的相异度(或相似度)是基于对象间的距离来计算的。

最常用的距离度量方法是欧几里得距离,它的定义如下:

$$d(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\cdots+|x_{ip}-x_{jp}|^2} \tag{9-7}$$

其中，$i=(x_{i1},x_{i2},\cdots,x_{ip})$和 $j=(x_{j1},x_{j2},\cdots,x_{jp})$是两个 p 维的数据对象。

另一个著名的度量方法是曼哈坦距离，其定义如下：

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\cdots+|x_{ip}-x_{jp}| \tag{9-8}$$

如果对每个变量根据其重要性赋予一个权重，加权的欧几里得距离可以计算如下：

$$d(i,j)=\sqrt{w_1|x_{i1}-x_{j1}|^2+w_2|x_{i2}-x_{j2}|^2+\cdots+w_p|x_{ip}-x_{jp}|^2} \tag{9-9}$$

加权也可用于曼哈坦距离。

9.2.3 二元变量及其相异度计算

1. 二元变量

一个二元变量只有 0 或 1 两个状态。0 表示该变量为空，1 表示该变量存在。例如，给出一个描述病人的变量 smoker，1 表示病人抽烟，而 0 表示病人不抽烟。

2. 相异度计算

如果假设所有的二元变量有相同的权重，得到如表 9-1 的一个表格。在表中，q 是对于对象 i 和 j 都为 1 的变量数目，r 是对于对象 i 值为 1 而对象 j 值为 0 的变量数目，s 是对于对象 i 值为 0 而对于对象 j 值为 1 的变量数目，t 是对于对象 i 和 j 值都为 0 的变量数目。变量的总数是 p，$p=q+r+s+t$。

表 9-1　二元变量的可能性表

	对象 j			
对象 i		1	0	求和
	1	q	r	$q+r$
	0	s	t	$s+t$
	求和	$q+s$	$r+t$	p

对于一个二元变量，如果它的两个状态有相同的权重，那么该二元变量是对称的，也就是两个取值 0 或 1 没有优先权，例如性别变量。基于对称二元变量的相似度称为恒定的相似度，即当一些或者全部二元变量编码改变时，计算结果不会发生变化。对恒定相似度来说，评价两个对象 i 和 j 之间相异度的最著名的系数是简单匹配系数，定义如下：

$$d(i,j)=\frac{r+s}{q+r+s+t} \tag{9-10}$$

如果两个状态的输出不是同样重要，那么该二元变量是不对称的。例如一个疾病检查的肯定和否定结果。根据惯例，将比较重要的输出结果，通常也是出现几率较小的结果编码为 1(例如，患病)，而将另一种结果编码为 0(例如，没病)。给定两个不对称的二元变量，两个都取值 1 的情况(正匹配)被认为比两个都取值 0 的情况(负匹配)更有意义。基于这样变量的相似度被称为非恒定的相似度。对于非恒定的相似度，最著名的评价系数是 Jaccard 系数，在计算中，负匹配的数目 t 被认为不重要的，因此被忽略。

$$d(i,j)=\frac{r+s}{q+r+s} \tag{9-11}$$

当对称和非对称的二元变量出现在同一个数据集中，可以应用混合变量方法。举例说明如下。

假设一个如表 9-2 所示的病人记录表包含属性姓名，性别，发烧，咳嗽，有测试-1，测试-2，测试-3 和测试-4 数据。

表 9-2　病人记录属性的关系表

姓名	性别	发烧	咳嗽	测试-1	测试-2	测试-3	测试-4
张三	男	Y	N	P	N	N	N
王五	女	Y	N	P	N	P	N
李四	男	Y	Y	N	N	N	N
…	…	…	…	…	…	…	…

"姓名"是对象标识，"性别"是对称的二元变量，其余的属性都是非对称的二元变量。对于非对称属性，值 Y 和 P 被置为 1，值 N 被置为 0。假设对象(病人)之间的距离只基于非对称变量来计算。根据 Jaccard 系数公式，三个病人两两之间的相异度如下：

$$d(\text{张三},\text{王五}) = \frac{0+1}{2+0+1} = 0.33$$

$$d(\text{张三},\text{李四}) = \frac{1+1}{1+1+1} = 0.67$$

$$d(\text{李四},\text{王五}) = \frac{1+2}{1+1+2} = 0.75$$

上面的值显示李四和王五不可能有相似的疾病，因为他们有着最高的相异度。在这 3 个病人中，张三和王五最可能有类似的疾病。

9.2.4　标称型变量及其相异度计算

1. 标称型变量

标称变量是二元变量的推广，它可以具有多于两个的状态值。例如，颜色是一个标称变量，它可能有 5 个状态：红色、蓝色、黄色、绿色和紫色。

假设一个标称变量的状态数目是 M。这些状态可以用字母、符号或者一组整数(如 0，1，2，…，M)来表示。要说明的是，这里的整数只是代表不同状态，没有任何顺序含义。

2. 相异度计算

两个对象 i 和 j 之间的相异度可用简单匹配方法来计算：

$$d(i,j) = \frac{p-m}{p} \tag{9-12}$$

其中，m 是匹配的数目，即 i 和 j 取值相同的变量的数目，而 p 是全部变量的数目。可以通过赋权重来增加 m 的影响，或者赋给有较多状态的变量匹配以更大的权重。

此外，通过为每个标称状态创建一个新的二元变量，可以用非对称的二元变量来编码标称变量。对一个有特定状态的对象，对应该状态值的二元变量值为 1，其余的二元变量值为 0。例如，为了对颜色变量进行编码，对应于上面的 5 种颜色分别创建一个二元变量。如果一个对象是黄色，那么 yellow 变量被赋值为 1，而其余的变量赋值为 0。对这种形式的标称变量编码，可以用计算二元变量相异度的方法来计算。

9.2.5 序数型变量及其相异度计算

1. 序数型变量

一个离散的序数型变量类似于标称变量，与标称标量的区别是序数型变量的 M 个状态是以有意义的序列排序的，而标称变量没有任何顺序含义。序数型变量对记录那些难以客观度量的主观评价是非常有用的。例如，职称的排列经常按某个顺序，例如教授、副教授、讲师、助教。

一个连续序数型变量看起来像一个刻度未知的连续数据的集合，也就是说，值的相对顺序是重要的，而其实际大小是不重要的。

将区间标度变量的值域划分为有限个区间，从而将其值离散化，也可得到序数型变量。一个序数型变量的值可以映射为秩(order)。例如，假设一个序数型变量有 M_f 个状态，这些有序的状态定义了一个排列 $1,2,\cdots,M_f$。

2. 相异度计算

计算对象的相异度时，序数型变量的处理与区间标度变量的处理非常类似。序数型对象间的相异度计算包括如下步骤。

(1) 设第 i 个对象的值为 x_{if}，且对应的序数型变量有 M_f 个有序的状态，对应于序列 $1,2,\cdots,M_f$。用对应的秩 r_{if} 代替 x_{if}，$r_{if}\in\{1,\cdots,M_f\}$。

(2) 既然每个序数型变量可以有不同数目的状态，必须将每个变量的值映射到[0.0, 1.0]上，以便每个变量都有相同的权重。这一点可以通过用 z_{if} 代替 r_{if} 来实现：

$$z_{if}=\frac{r_{if}-1}{M_f-1} \tag{9-13}$$

(3) 相异度的计算可采用一种距离度量方法，采用 z_{if} 作为第 i 个对象的 x_{if} 值。

9.2.6 比例标度型变量及其相异度计算

1. 比例标度型变量

比例标度型变量总是取正的度量值，有一个非线性的标度，近似的遵循指数标度，如

$$A\mathrm{e}^{BT} \quad \text{或} \quad A\mathrm{e}^{-BT}$$

其中，A 和 B 是正的常数。典型的例子包括细菌数目的增长，或者放射性元素的衰变。

2. 相异度计算

对于这种变量对应的对象的相异度计算，目前有 3 种方法。

(1) 采用与区间标度变量同样的方法。但这不是一个好的选择，因为标度可能被扭曲了。

(2) 将比例标度型变量进行对数变换，例如对象 i 的值 x_{if} 被变换为 y_{if}，$y_{if}=\log(x_{if})$。变换得到的 y_{if} 值可以采用区间标度变量描述的方法处理。

(3) 将 x_{if} 看作连续的序数型数据，将其秩作为区间标度的值来对待。

9.2.7 混合类型变量的相异度计算

前面所讨论的都是由相同类型变量描述的对象之间的相异度计算方法，变量的类型可能是区间标度变量、对称二元变量、不对称二元变量、标称变量、序数型变量或者比例标度变

量。但在许多真实的数据库中,对象是被混合型的变量描述的。一般来说,一个数据库可能包含上面列出的全部 6 种变量类型。

利用混合类型变量来描述对象之间的相异度,一种方法是将变量按类型分组,对每种类型的变量进行单独的聚类分析。如果这些分析得到兼容的结果,这种方法是可行的。但在实际中,这种方法的实用性不大。

一个更可取的方法是将所有的变量一起处理,只进行一次聚类分析。一种技术将不同类型的变量组合在单个相异度矩阵中,把所有有意义的变量转换到共同的域值[0.0,1.0]上。

假设数据集包含 p 个不同类型的变量,对象 i 和对象 j 之间的相异度 $d(i,j)$ 定义为

$$d(i,j)=\frac{\sum_{f=1}^{p}\delta_{ij}^{(f)}d_{ij}^{(f)}}{\sum_{f=1}^{p}\delta_{ij}^{(f)}} \tag{9-14}$$

其中,如果 x_{if} 或 x_{jf} 缺失(即对象 i 或对象 j 没有变量 f 的度量值),或者 $x_{if}=x_{jf}=0$,且变量 f 是不对称的二元变量,则指示项 $\delta_{ij}^{(f)}=0$;否则 $\delta_{ij}^{(f)}=1$。变量 f 对对象 i 和对象 j 之间相异度的计算方式与其具体类型有关。

(1) 当 f 是二元变量或标称变量时:如果 $x_{if}=x_{jf}$,$d_{ij}^{(f)}=0$;否则 $d_{ij}^{(f)}=1$。

(2) 当 f 是区间标度变量时:$d_{ij}^{(f)}=\frac{|x_{if}-x_{jf}|}{\max_h x_{hf}-\min_h x_{hf}}$,这里的 h 取遍变量 f 的所有非空缺对象。

(3) 当 f 是序数型或者比例标度型变量时:计算秩 r_{if} 和 $z_{if}=\frac{r_{if}-1}{M_f-1}$,并将 z_{if} 作为区间标度变量值对待。

这样,当描述对象的变量类型是不同类型时,对象之间的相异度也能够进行计算。

9.3 基于划分的聚类方法

给定一个 n 个对象或元组的数据源,划分方法将数据构建为 k 个划分,每个划分表示一个聚簇,并且 $k\leqslant n$。也就是说,它将数据划分为 k 个组,同时满足如下要求:

(1) 每个组至少包含一个对象;

(2) 每个对象必须属于一个组。

在某些模糊划分技术中第二个要求可以放宽。

常用的划分方法有 k-平均算法和 k-中心点算法两种。

9.3.1 k-平均算法

在该算法中,每个簇用该簇中对象的平均值来表示。k-平均算法不适合于处理分类属性数据;对数值数据有较好的几何和统计意义。

k-平均算法的核心思想是通过迭代把数据对象户划分到不同的簇中,以求目标函数最小化,从而使生成的簇尽可能地紧凑和独立。

具体划分过程是,首先,随机选取 k 个对象作为初识的 k 个簇的质心;然后,将其余对象

根据其与各个簇质心的距离分配到最近的簇;最后,再求新形成的簇的质心。如此迭代、重定位,尝试通过对象在划分之间移动来改进划分。一个好的划分的一般规则是:在同一个簇中的对象之间尽可能“接近”或相关,而不同簇中的对象之间尽可能“远离”或不同。

k-平均算法的处理流程:首先,随机选择 k 个对象,每个对象初始地代表了一个簇的平均值或中心。对剩余的对象,根据其与各个簇中心的距离,将它赋给最近的簇。然后重新计算每个簇的平均值。这个过程不断重复,直到准则函数收敛。通常采用平方误差准则,其定义如式:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} | p - m_i |^2 \tag{9-15}$$

其中,E 是数据库中所有对象平方误差的总和,p 是空间中的点,表示给定的数据对象,m_i 是簇 C_i 的平均值(p 和 m_i 都是多维的)。这个准则试图使生成的结果簇尽可能地紧凑和独立。

k-平均算法尝试找出使平方误差函数值最小的 k 个划分。当结果簇是密集的,并且簇与簇之间区分明显时,它的效果较好。对处理大数据集,该算法是相对可伸缩和高效率的,因为它的复杂度是 $O(nkt)$,其中,n 是所有对象的数目,k 是簇的数目,t 是迭代的次数。通常地,$k << n$,且 $t << n$。

k-平均算法具有以下不足:

(1) k-平均算法经常以局部最优结束。

(2) k-平均算法只有在簇的平均值被定义的情况下才能使用,这可能不适用于某些应用,例如涉及有分类属性的数据。

(3) 要求用户必须事先给出 k(要生成的簇的数目)也可以算是方法的一个缺点。

(4) k-平均算法不适合于发现非凸面形状的簇,或者大小差别很大的簇。

(5) k-平均算法对于“噪声”和孤立点数据是敏感的,少量的该类数据能够对平均值产生极大的影响。

9.3.2 *k*-中心点算法

k-中心点算法选择簇中位置最接近聚中心的对象作为簇的代表点。该算法的处理过程是:首先,随机选择 k 个对象作为初始的 k 个簇的代表点,将其余对象根据其与代表点对象的距离分配到最近的簇;然后,反复用非代表点来取代代表点,以改进聚类质量。聚类质量用代价函数来估计,该函数度量对象与代表点对象之间的平均相异度。

k-中心点算法对属性类型没有局限性,通过簇内主要点的位置来确定选择中心点,对孤立点的敏感性小。

划分聚类方法对在中小规模的数据集中发现球状簇很适用。为了对大规模的数据集进行聚类,以及处理复杂形状的聚类,基于划分的方法需要进一步的扩展。

k-平均算法对于孤立点敏感,为消除这种敏感性,不采用簇中对象平均值作为参考点,而选用簇中位置最中心的对象,即中心点为参考点,这就是 k-中心方法。

聚类结果质量用一个代价函数来估算,该函数度量对象与其参照对象之间的平均相异度。为了判定一个非代表对象 Orandom 是否是当前一个代表对象 O_j 的好的替代,对每个非中心点对象 p,考虑下面的 4 种情况。

(1) 当前隶属于中心对象 O_j。

如果 O_j 被 O_{random} 所代替作为中心点，且 p 离一个 O_i 最近，$i \neq j$，那么 p 被重新分配给 O_i。

(2) 当前隶属于中心对象 O_j。

如果 O_j 被 O_{random} 所代替作为中心点，且 p 离 O_{random} 最近，那么 p 被重新分配给 O_{random}。

(3) 当前隶属于中心点 O_i，$i \neq j$。

如果 O_j 被 O_{random} 所代替作为中心点，而 p 仍然离 O_i 最近，那么对象的隶属不发生变化。

(4) 当前隶属于中心点 O_i，$i \neq j$。

如果 O_j 被 O_{random} 所代替作为中心点，而 p 离 O_{random} 最近，那么 p 被重新分配给 O_{random}。

一个典型的 k-中心点算法描述如下。

算法：基于中心点或中心对象划分的典型 k-中心点算法。

输入：簇的数目 k 和包含 n 个对象的数据库。

输出：k 个簇，使所有对象与其最近中心点的相异度综合最小。

方法：

(1) 选择 k 个对象作为初始的中心点；

(2) repeat；

(3) 指派每个剩余的对象给离它最近的中心点所代表的簇；

(4) 随机的选择一个非中心点对象 O_{random}；

(5) 计算用 O_{random} 代替 O_j 的总代价 S；

(6) if $S<0$ then O_{random} 替换 O_j，形成新的 k 个中心点的集合；

(7) until 不发生变化。

当存在"噪声"孤立点数据时，k-中心点方法比 k-平均方法更健壮，这是因为中心点不像平均值那么容易被极端数据影响。但是，k-中心点方法的执行代价比 k-平均方法高。此外这两种方法都要求用户指定结果簇的数目 k。

9.4 基于层次的聚类方法

基于层次的聚类方法对给定数据对象集合进行层次的分解。根据层次分解形成的顺序，层次的方法可以分为凝聚的方法和分裂的方法。

1. 凝聚的方法

凝聚的方法，也称为自底向上的方法，一开始将每个对象作为单独的一个组，然后相继地合并相近的对象或组，直到所有的组合并为一个(层次的最上层)，或者达到一个终止条件。

2. 分裂的方法

分裂的方法，也称为自顶向下的方法，一开始将所有的对象置于一个簇中，在迭代的每一步中，一个簇被分裂为更小的簇，直到最终每个对象在单独的一个簇中，或者达到一个终止条件。

图 9-1 中描述了一个聚类的层次聚类方法 AGNES(agglomerative nesting)和一个分裂的层次聚类方法 DIANA(divisive analysis)在一个包含 5 个对象的数据集合$\{a,b,c,d,e\}$上

的处理过程。

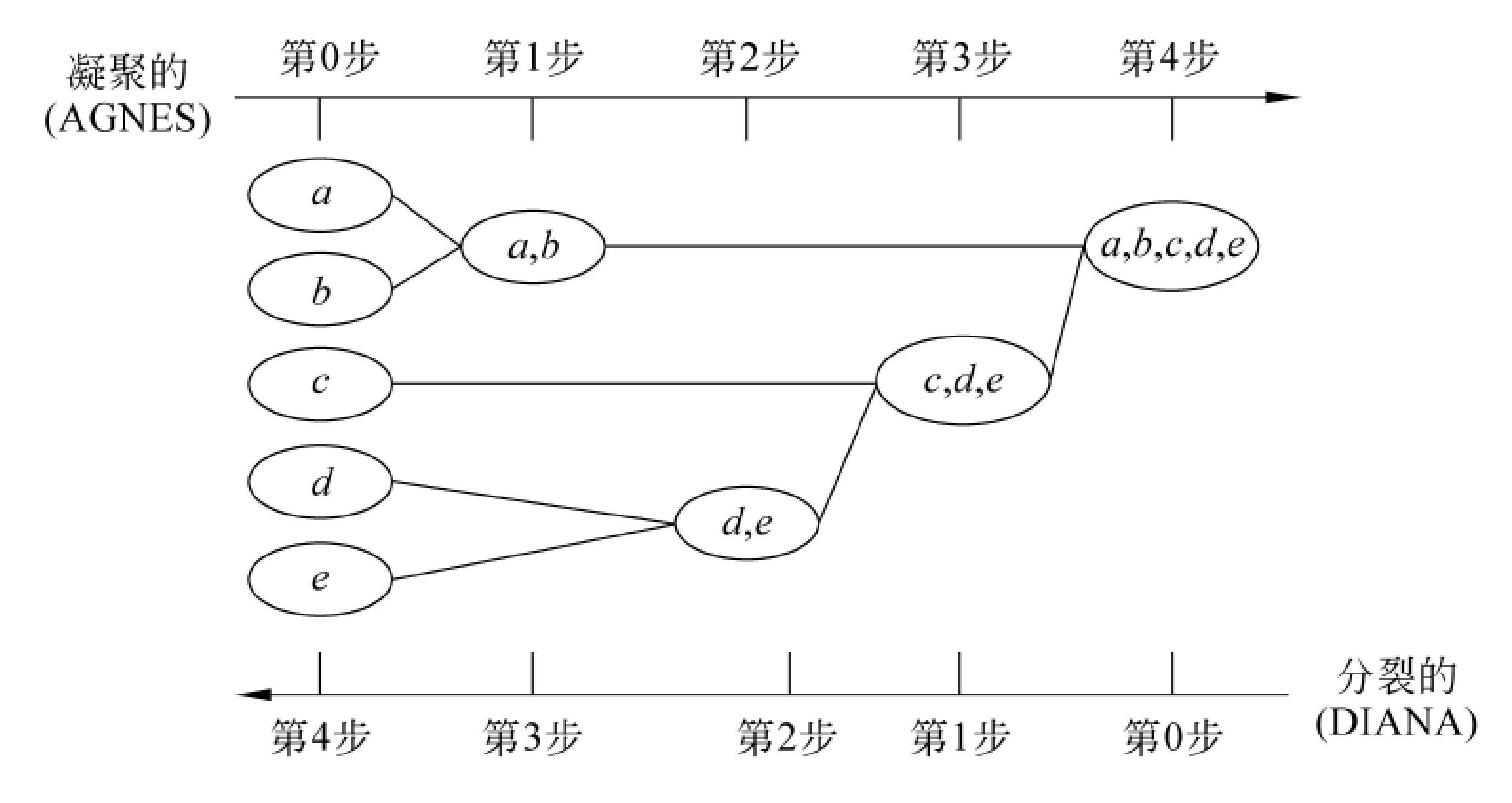


图 9-1 在数据集{a,b,c,d,e}上的凝聚和分裂层次聚类

最初,AGNES 将每个对象作为一个簇,然后这些簇根据某些准则被一步步合并。例如,如果簇 C_1 中的一个对象和簇 C_2 中一个对象之间的距离是所有属于不同簇的对象间欧几里得距离中最小的,C_1 和 C_2 可能被合并。这是一种单链接(single-link)方法,其每个簇可以被簇中所有对象代表,两个簇间的相似度由这两个不同簇中距离最近的数据点对的相似度来确定。聚类的合并过程反复进行直到所有的对象最终合并形成一个簇。

在 DIANA 方法的处理过程中,所有的对象开始都放在一个簇中。根据一些原则(如簇中最临近对象的最大欧几里得距离),将该簇分裂。簇的分裂过程反复进行,直到最终每个新的簇只包含一个对象。

在凝聚或者分裂的层次聚类方法中,用户定义希望得到的簇数目作为一个结束条件。

层次聚类方法尽管简单,但经常会遇到合并或者分裂点选择的困难,这样的决定是非常关键的,因为一旦一组对象被合并或者分裂,下一步的处理将在新生成的簇上进行。已做的处理不能被撤销,聚类之间也不能交换对象。如果在某一步没有很好的选择合并或分裂的决定,可能会导致低质量的聚类结果。而且,这种聚类方法不具有很好的可伸缩性,因为合并或分裂的决定需要检查和估算大量的对象或簇。

9.5 谱聚类方法

谱聚类(spectral clustering)算法建立在图论中的谱图理论基础上,其本质是将聚类问题转化为图的最优划分问题,是一种点对聚类算法。与传统的聚类算法相比,它具有能在任意形状的样本空间上聚类且收敛于全局最优解的优点。

9.5.1 谱聚类的步骤

(1) 根据数据构造一个 Graph,Graph 的每一个结点对应一个数据点,将相似的点连接起来,并且边的权重用于表示数据之间的相似度。把这个 Graph 用邻接矩阵的形式表示出来,记为 $\boldsymbol{W}$。

(2) 把 $\boldsymbol{L}$ 的每一列元素加起来得到 N 个数,把它们放在对角线上(其他地方都是零),

组成一个 $N \times N$ 的矩阵，记为 $\boldsymbol{D}$。并令 $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$。

(3) 求出 L 的前 k 个特征值（"前 k 个"指将特征值按照从小到大的顺序排列后的前 k 个）以及对应的特征向量。

(4) 把这 k 个特征(列)向量排列在一起组成一个 $N \times k$ 的矩阵，将其中每一行看作 k 维空间中的一个向量，并使用 K-means 算法进行聚类。聚类的结果中每一行所属的类别就是原来 Graph 中的结点亦即最初的 N 个数据点分别所属的类别。

9.5.2 谱聚类的优点

谱聚类方法和传统聚类方法(例如 k-means)比起来有不少优点。

(1) 谱聚类方法只需要数据之间的相似度矩阵信息，而不必像 k-means 那样要求数据必须是 N 维欧几里得空间中的向量。

(2) 由于抓住了主要矛盾，因此比传统的聚类算法更加健壮，对于不规则的误差数据不是特别敏感，而且性能也要好一些。许多实验都证明了这一点。

(3) 计算复杂度比 k-means 算法要小，特别是在像文本数据或者平凡的图像数据这样维度非常高的数据上运行的时候。

9.5.3 谱聚类实例

设有 3 个对象分别记为对象 1，对象 2，对象 3，3 个对象的相似度矩阵为

$$\boldsymbol{D}=\begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

矩阵 $\boldsymbol{D}$ 每列的元素相加得到对角矩阵

$$\boldsymbol{W}=\begin{bmatrix}2 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 1\end{bmatrix}$$

矩阵

$$\boldsymbol{D}-\boldsymbol{W}=\begin{bmatrix}1 & -1 & 0\\ -1 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}$$

矩阵

$$\boldsymbol{D}-\boldsymbol{W}=\begin{bmatrix}1 & -1 & 0\\ -1 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}$$

的特征值为 0，0，2，

特征向量为

$$\begin{bmatrix}-0.7071\\ -0.7071\\ 0\end{bmatrix},\begin{bmatrix}0\\ 0\\ 1\end{bmatrix},\begin{bmatrix}-0.7071\\ 0.7071\\ 0\end{bmatrix}$$

在谱聚类中，一般根据等于 0 或接近于 0 的特征值的个数来确定聚类数目。根据这个

原则,这三个对象应该聚成两类。

取这两个特征值对应的特征向量构成矩阵

$$\begin{bmatrix} -0.7071 & 0 \\ -0.7071 & 0 \\ 0 & 1 \end{bmatrix}$$

将这个矩阵的每行看成一个点,共得到 3 个点 $a=(-0.7071\quad 0)$,$b=(-0.7071\quad 0)$,$c=(0\quad 1)$。

上述 3 个点显然点 a 和点 b 聚成一类,点 c 自己聚成一类。对应的原来 3 个对象中,对象 1 和对象 2 聚成一类,对象 3 自己聚成一类。

9.6 利用 SQL Server 2005 进行聚类分析

9.6.1 挖掘流程

聚类是一个强大的工具,可以按照相似性对数据进行分组;可以用来理解数据;也可以用来作为数据分析的一个关键步骤。UCI 数据源中的 Iris 数据集以鸢尾花的特征作为数据来源,数据集包含 150 个样本,每类 50 个样本,每个样本包含 4 个属性值,分别为花瓣长度、花瓣宽度、花萼长度、花萼宽度。在此利用 SQL Server 2005 对 Iris 数据进行聚类分析。

(1) 右击项目 Clustering Analysis 下的"挖掘结构"选择"新建挖掘结构",打开"数据挖掘向导"对话框,单击"下一步"按钮,进入"选择定义方法"页面,单击"下一步"按钮,进入"选择数据挖掘技术"对话框。

(2) 在如图 9-2 所示的对话框中,下拉列表框中选取"Microsoft 聚类分析"选项,单击"下一步"按钮,进行下一步操作。

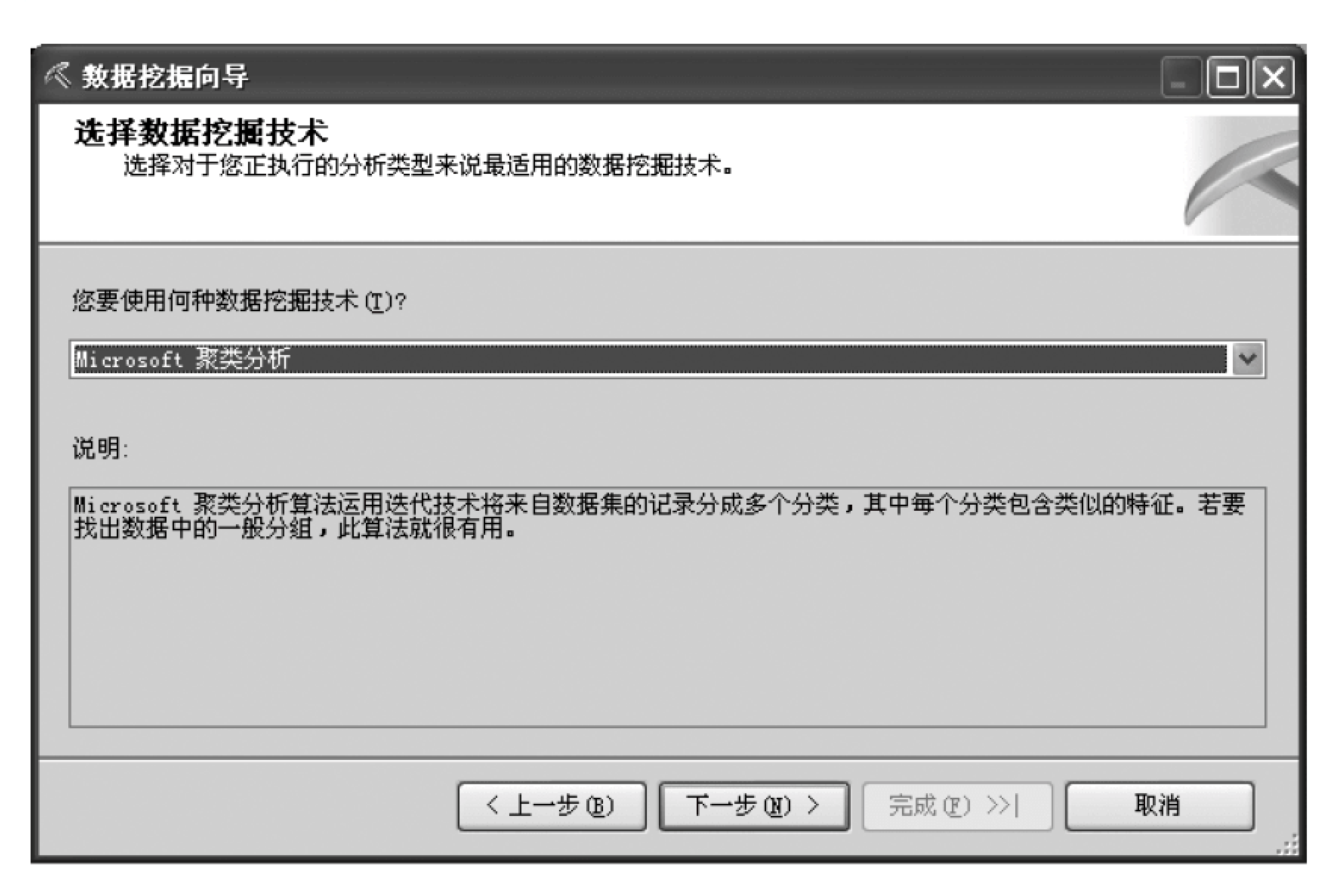


图 9-2 选择数据挖掘技术

(3) 如图 9-3 所示,在"选择数据源视图"页面的"可用数据源视图"列表中显示了前面步骤创建的 Iris 数据源视图,选中该视图选项,单击"下一步"按钮,进行下一步操作。



图 9-3　选择数据源视图

(4) 如图 9-4 所示，在"指定表类型"页面中可以看到 Iris 数据源视图包含的数据表，勾选"Sheet1 $ "选项右边的"事例"复选框，可以将其定义为事例表；单击"下一步"按钮，进行下一步操作。



图 9-4　指定表类型

(5) 如图 9-5 所示，在"指定定型数据"页面显示了挖掘模型结构，在各个选项右边勾选不同的复选框，然后单击"下一步"按钮，进行下一步操作。

(6) 如图 9-6 所示，在"指定列的内容和数据类型"对话框中显示了指定"ID"的内容类型为 Key，Class 的内容类型为 Discrete，其余列内容类型均为 Continuous；Class 的数据类型为 Text，各其余列数据类型均为 Double，单击"下一步"按钮，进行下一步操作。

(7) 如图 9-7 所示，在"完成向导"页面中将数据挖掘结构命名为 iris，单击"完成"按钮，完成挖掘结构的创建。

9.6.2　结果分析

单击"挖掘模型查看器"下的"分类关系图"、"分类剖面图"、"分类特征"、"分类对比"选项卡，可以得到如图 9-8 所示的结果。



图 9-5　指定定型数据



图 9-6　指定列的内容和数据类型

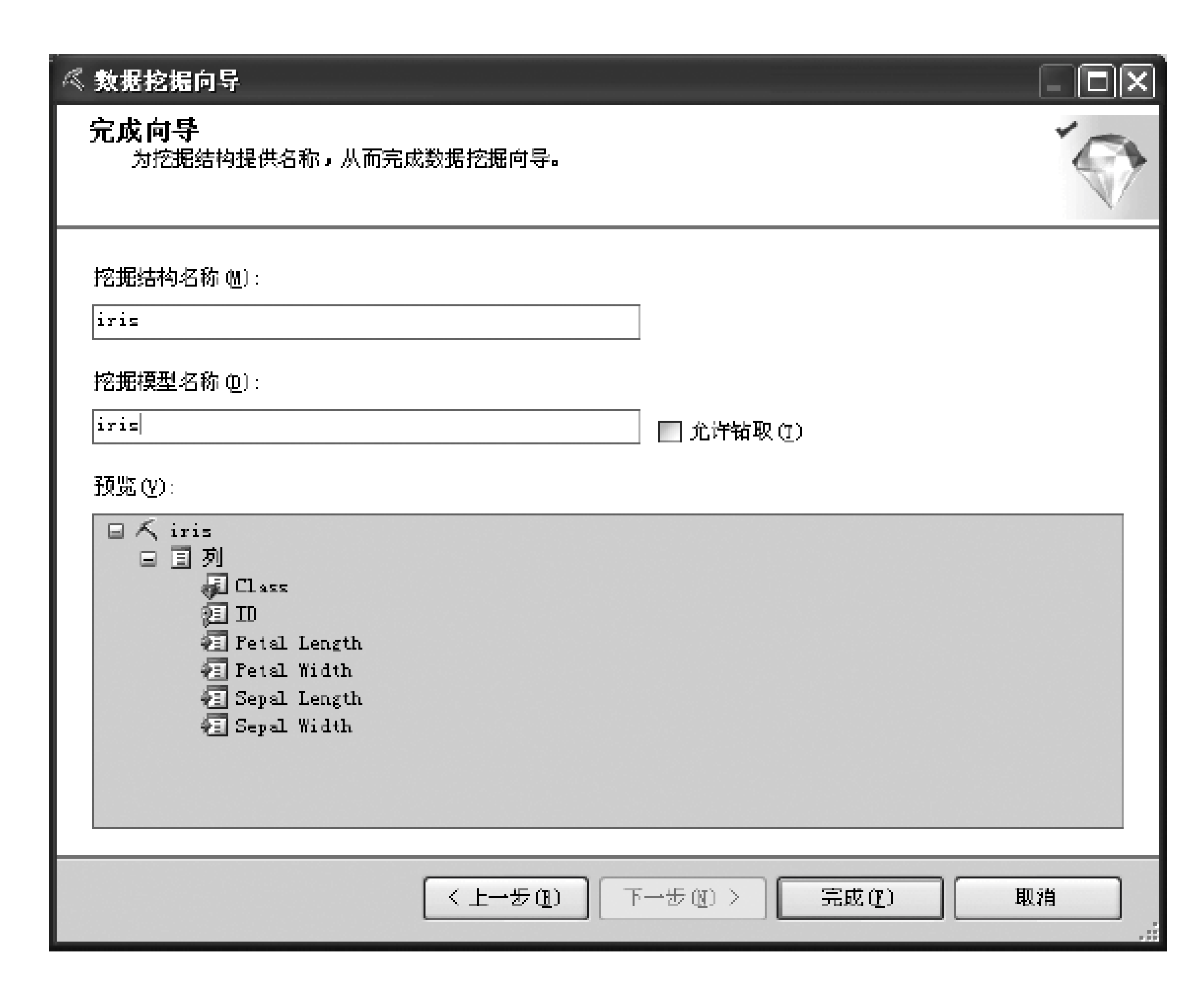


图 9-7　完成数据挖掘结构的创建



图 9-8　分类剖面图

在图 9-8 中的分类剖面视图中每一列对应模型中的每一个聚类，每一行对应于一个属性，根据这样的设置，可以很容易地看出这些聚类之间的不同点，使用这个视图，可以选择一个您感兴趣的属性，并且可以可视地水平扫描来查看该属性在所有据类中的分布，如果观察该项相邻的单元或者同一聚类的其他单元，将会发现有关该聚类含义的更多信息。

在图 9-9 中的分类关系视图是聚类查看器的第一个选项卡，也是聚类查看器的默认选项卡，在视图中，每一个聚类用一个结点表示，这些结点是分散在某个区域中的，可以基于这些聚类的相似性对它们进行分组。该视图以图的方式显示哪一些聚类相似或者不相似，并且显示了它们的相似性的相对程度。

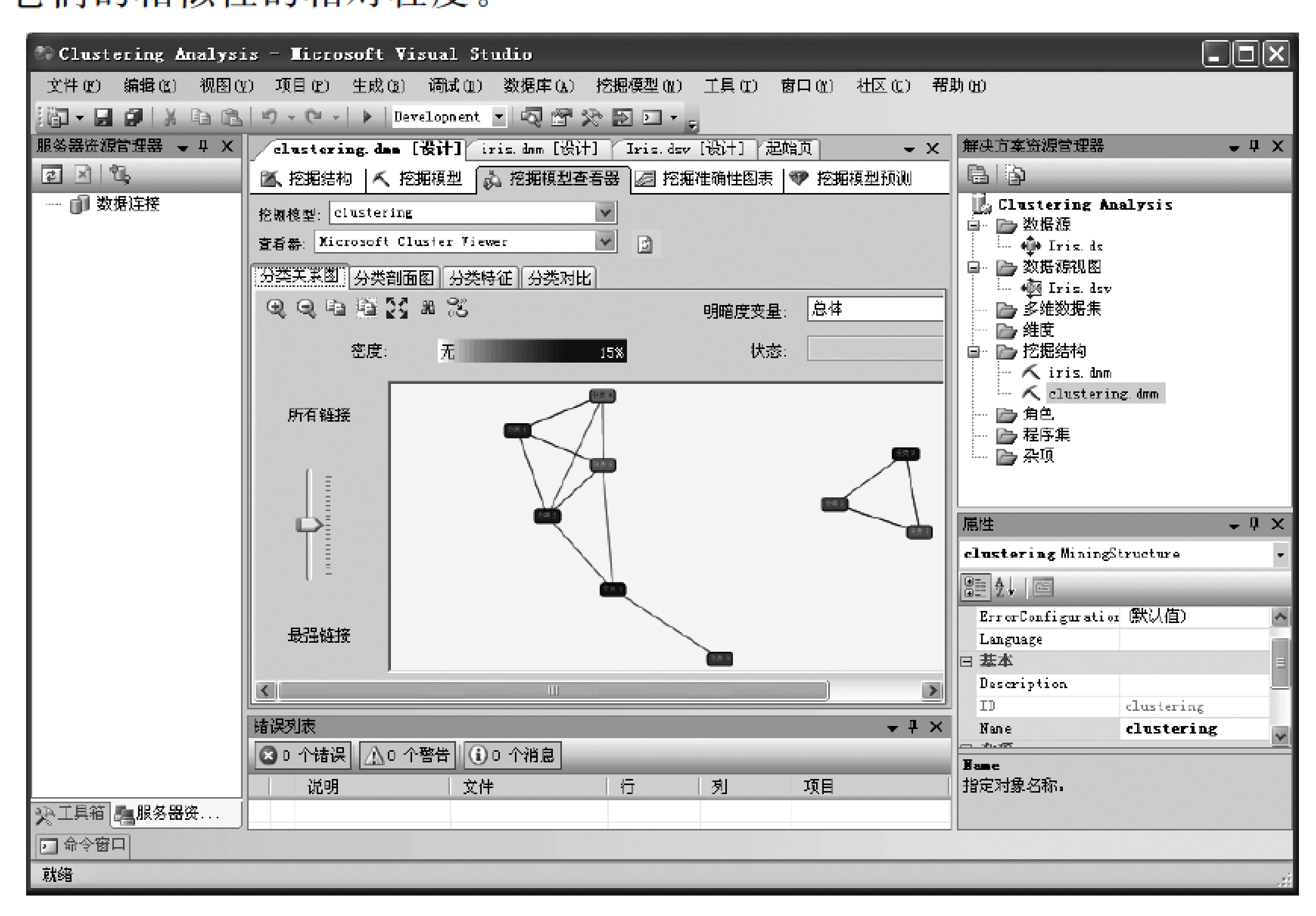


图 9-9　分类关系图

在图 9-10 分类特征视图中，主要描述的是所选聚类事例的特征，通过递减概率显示属性来描述该特征。通过将某一聚类与其他聚类进行比较来确定对于该聚类中什么属性是最重要的，如图 9-11 所示。



图 9-10 分类特征

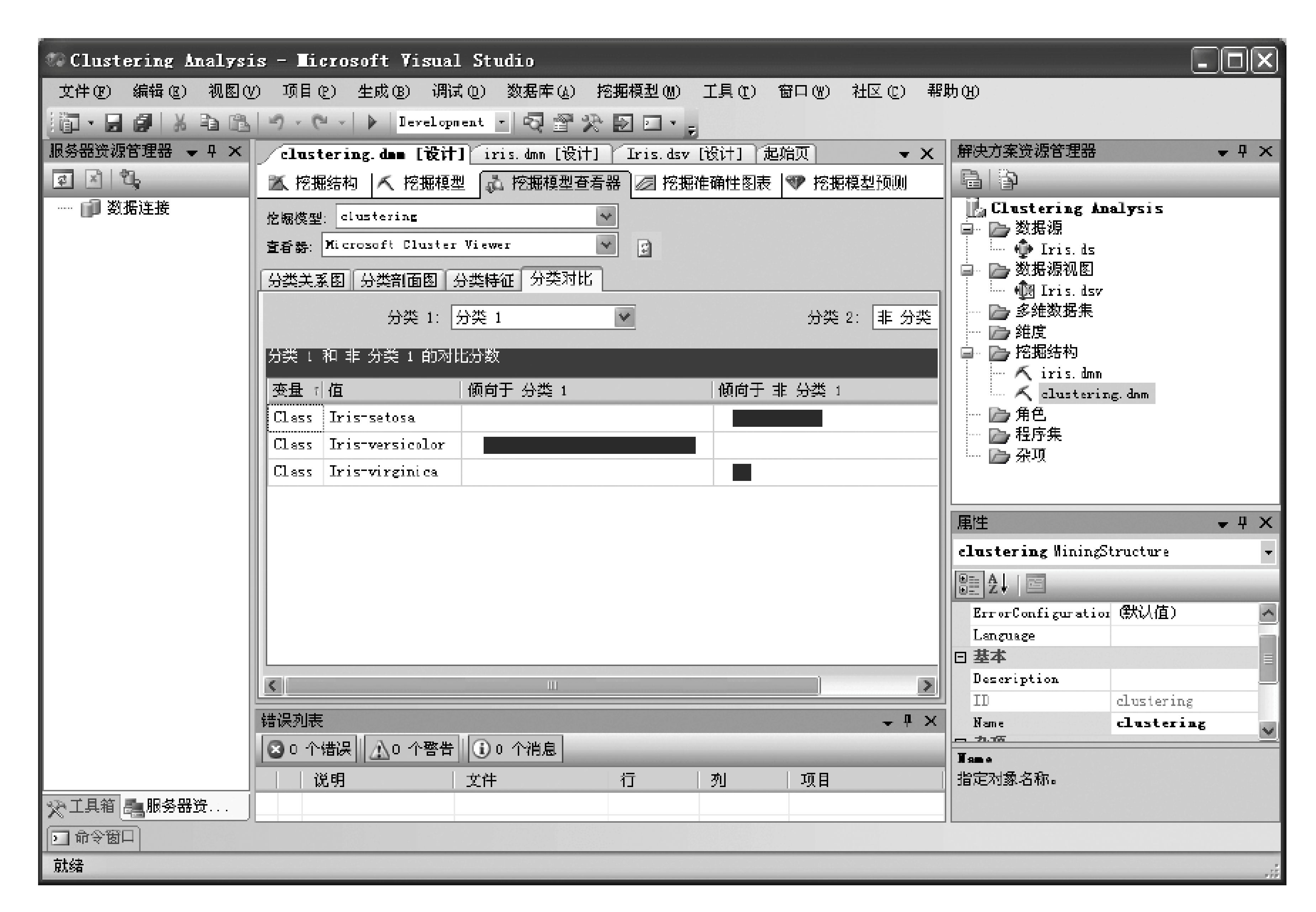


图 9-11 分类对比

单击“挖掘准确性图表”选项卡下的“提升图”和“分类矩阵”，其结果如图 9-12 所示。

通过图 9-13 可以看出，150 个数据中，一共有 50+47+48=145 个数据被正确地进行了聚类，只有 5 个数据没有被正确地进行聚类，总体上得到了比较满意的聚类结果。

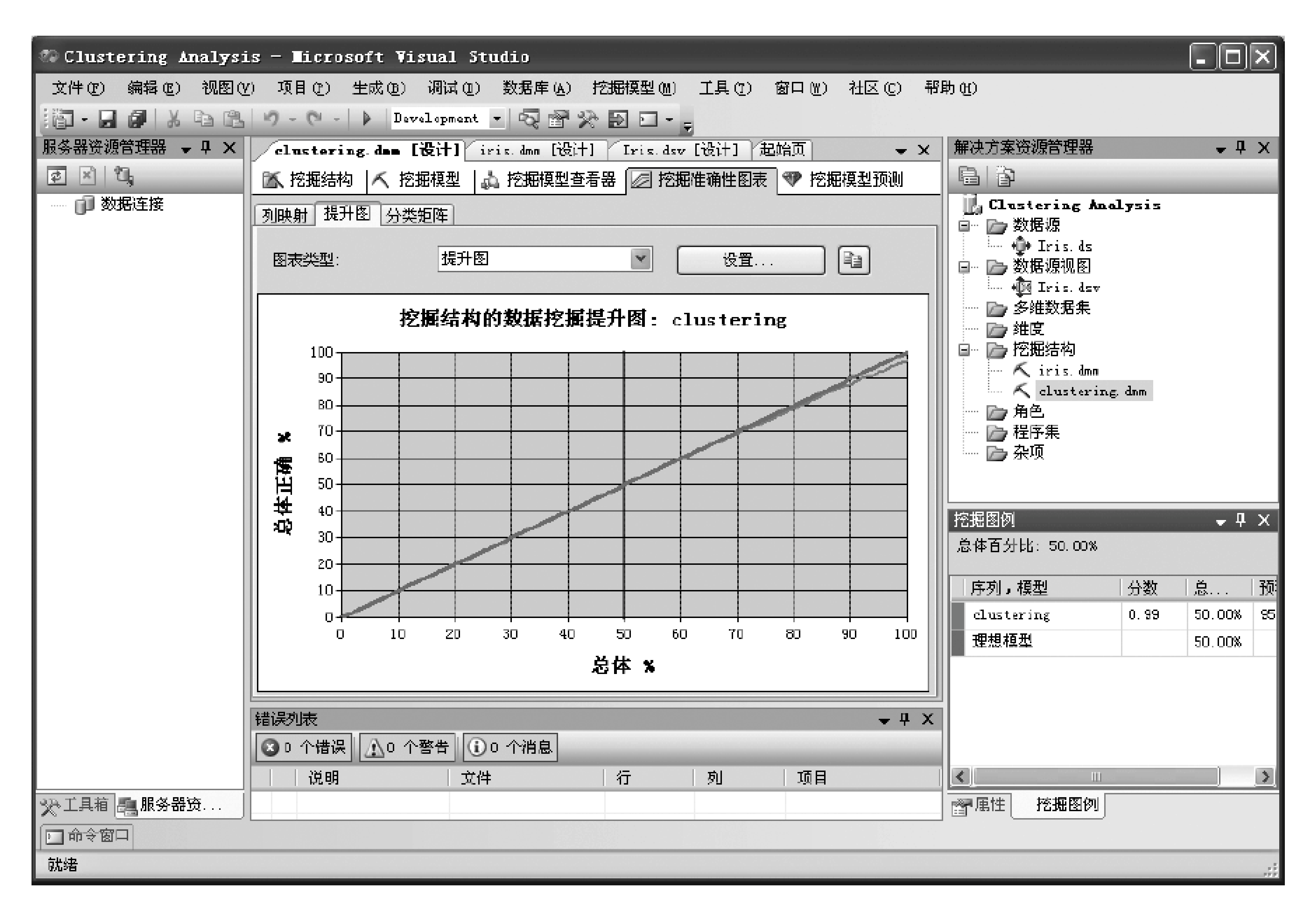


图 9-12　提升图



图 9-13　分类矩阵图

小结

本章首先介绍了聚类分析的概念以及要求。然后分不同变量类型介绍了距离的度量方法，包括区间标度变量、二元变量和标称型变量等。接下来介绍了基于划分的聚类方法和基

于层次的聚类方法，其中基于划分的聚类方法包括 k-平均算法和 k-中心点算法，基于层次的聚类方法包括凝聚算法和分裂算法，最后介绍了基于 SQL Server 2005 的聚类分析实现技术，使读者进一步体会了如何利用 SQL Server 2005 具体实现聚类模式的挖掘。

习题 9

1. 简单地描述如何计算由如下类型的变量描述的对象间的相异度：

(a) 数值(区间标度)变量

(b) 非对称的二元变量

(c) 分类变量

(d) 比例标度变量

(e) 非数值向量对象

2. 假设数据挖掘的任务是将如下 8 个点聚类为 3 个簇：

A1(2,10),A2(2,5),A3(8,4),B1(5,8),B2(7,5),B3(6,4),C1(1,2),C3(4,9),距离函数是欧几里得距离。假设初始选择 A1,B1,C1 分别为每个聚类的中心，用 k-平均算法来给出：

(1) 在第一次循环执行后的三个聚类中心。

(2) 最后的三个簇。

3. k 均值和 k 中心点算法都可以进行有效的聚类。概述 k 均值和 k 中心点算法的优缺点。并概述这两种方法与层次聚类方法(如 AGNES)相比有何优缺点。

第 10 章　粗糙集方法

粗糙集(rough set,RS)是一种处理不确定、不完备数据和不精确问题的新的数学理论,最初由波兰数学家 Z. Pawlak 于 1982 年提出。粗糙集理论建立在分类机制的基础上,将知识理解为对数据的划分,并引入上近似和下近似等概念来刻画知识的不确定性和模糊性。近年来,它已被广泛应用到人工智能、模式识别和数据挖掘等方面。

10.1　粗糙集的基本概念

10.1.1　等价关系与等价类

1. 笛卡儿乘积

集合 A,B 的笛卡儿乘积为 $A\times B$,它是二元组集合 $\{(a,b)|a\in A\wedge b\in B\}$,即

$$A\times B=\{(a,b)\mid a\in A\wedge b\in B\} \tag{10-1}$$

例如:设 $A=\{a,b,c\}$,$B=\{1,2\}$,则 $A\times B=\{(a,1),(a,2),(b,1),(b,2),(c,1),(c,2)\}$,$A\times A=\{(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)\}$。

2. 二元关系

$A\times B$ 的子集叫做 A 到 B 的一个二元关系,特殊地,$A\times A$ 的子集叫做 A 上的一个二元关系。

例如:设 $A=\{a,b,c\}$,$B=\{1,2\}$,则 $\{(a,1),(a,2),(b,1),(c,2)\}$ 为 A 到 B 的一个二元关系;$\{(a,a),(a,b),(a,c),(b,a),(b,b),(c,a),(c,c)\}$ 为 A 上的一个二元关系。

3. 等价关系

如果集合 A 上的二元关系 R 满足:

(1) 对任意的 $a\in A$,有 $(a,a)\in R$;

(2) 对任意的 $a,b\in A$,若 $(a,b)\in R$,则 $(b,a)\in R$;

(3) 对任意的 $a,b,c\in A$,若 $(a,b)\in R$,$(b,c)\in R$,则 $(a,c)\in R$。

则称 R 是 A 上的等价关系。

若 $(x,y)\in R$,则称 x 和 y 有关系,记为 xRy;若 $(x,y)\notin R$,则称 x 和 y 没有关系,记为 $x\bar{R}y$。

例如:设 $A=\{a,b,c,d,e,f\}$,可以验证 $R=\{(a,a),(b,b),(c,c),(d,d),(e,e),(f,f),(a,b),(b,a),(a,c),(c,a),(b,c),(c,b),(d,e),(e,d)\}$ 为 A 上的等价关系。

4. 等价类

设 R 是集合 A 上的等价关系,对每一 $a\in A$,a 关于 R 的等价类是集合 $\{x|xRa\}$,记为 $[a]_R$,简记为 $[a]$。

例如:设集合 $A=\{a,b,c,d,e,f\}$,A 上的一个等价关系 $R=\{(a,a),(b,b),(c,c),(d,d),(e,e),(f,f),(a,b),(b,a),(a,c),(c,a),(b,c),(c,b),(d,e),(e,d)\}$,则可得到各元素

的等价类如下：

$$[a]=[b]=[c]=\{a,b,c\},[d]=[e]=\{d,e\},[f]=\{f\}$$

5. 划分

给定非空集合 A 和非空集合簇 $\pi=\{A_1,A_2,A_m\}$，如果

（1）π 是 A 覆盖，即

$$A=\bigcup_{i=1}^{m} A_i \tag{10-2}$$

（2）

$$A_i \cap A_j = \varnothing \quad 或 \quad A_i = A_j (i,j=1,2,\cdots,m) \tag{10-3}$$

那么称集合簇 π 是 A 的一个划分。

不难得到以下结论：设 A 是非空集合，R 是 A 上的等价关系，则 R 的等价类集合$\{[a]_R \mid a\in A\}$是 A 的划分。

10.1.2 信息表与决策表

1. 信息表

一个信息表 S 定义为一个四元组：$S=\langle U,A,V,f\rangle$，其中 $U=\{u_1,u_2,\cdots,u_n\}$是对象集合，即论域；A 是属性集合；V 是属性值的集合；f 是一个信息函数，它指定 U 中每一个对象的属性值。

表 10-1 给出了一个包含 6 个对象，4 个属性的信息表。

表 10-1 一个信息表实例

个体编号	头疼	肌肉疼	体温	流感
e_1	是	是	正常	否
e_2	是	是	高	是
e_3	是	是	很高	是
e_4	否	是	正常	否
e_5	否	否	高	否
e_6	否	是	很高	是

2. 决策表

对信息表 S，若属性集可分为条件属性集 C 和决策属性集 D，即有 $A=C\cup D$，且 $C\cap D=\varnothing$（一个属性不能同时为条件属性和决策属性），则该信息表称为一个决策表 L。

表 10-2 给出了一个包含 5 个对象，2 个条件属性，1 个决策属性的决策表。

表 10-2 一个决策表实例

样例	c_1	c_2	d_1
1	a	1	+
2	b	3	+
3	a	1	−
4	c	2	−
5	b	2	+

10.1.3 下近似与上近似

1. 不可分辨关系

在信息表 S 中，对于属性集 $I \subseteq A$，可构造对应的二元等价关系 $\mathrm{IND}(I)=\{<x,y>\in U\times U \mid \forall a\in I$，有 $a(x)=a(y)\}$，其中 $a(x)$ 表示对象 x 在属性 a 上的属性值，称 $\mathrm{IND}(I)$ 为由 I 构造的不可分辨关系。

对于表 10-1 所示的信息表，按照头疼属性进行分类，可以得到划分如下：

$$U/\text{头疼} = \{\{e_1,e_2,e_3\},\{e_4,e_5,e_6\}\}$$

这里，e_1，e_2 和 e_3 这三个对象在头疼属性上是不可区分的，即它们一起构成一个等价类，e_4，e_5，e_6 这三个个体构成另一个等价类。

按照肌肉疼、体温和流感这三个属性分别进行分类，可以得到 3 个划分如下：

$$U/\text{肌肉疼} = \{\{e_1,e_2,e_3,e_4,e_6\},\{e_5\}\}$$

$$U/\text{体温} = \{\{e_1,e_4\},\{e_2,e_5\},\{e_3,e_6\}\}$$

$$U/\text{流感} = \{\{e_1,e_4,e_5\},\{e_2,e_3,e_6\}\}$$

按照头疼和肌肉疼这两个属性来共同分类，可以得到划分如下：

$$U/\text{头疼和肌肉疼} = \{\{e_1,e_2,e_3\},\{e_4,e_6\},\{e_5\}\}$$

2. 下近似与上近似

设 $X\subseteq U$ 为感兴趣的对象集合，A 为属性集合，则 X 的 A 下近似(lower approximation)定义为：

$$\underline{A}(X) = \bigcup \{Y \in U/\mathrm{A} \mid Y \subseteq X\} \tag{10-4}$$

它是一定属于 X 的所有对象的集合。其中 U/A 表示对象集合 U 关于属性集合 A 的等价类集合。

X 的 A 上近似(upper approximation)定义为：

$$\overline{A}(X) = \bigcup \{Y \in U/\mathrm{A} \mid Y \cap X \neq \varnothing\} \tag{10-5}$$

它是可能属于 X 的所有对象的集合。

在表 10-1 所示的决策表中，对于属性子集 $B=\{$头疼，肌肉疼$\}$，集合 $X=\{e_2,e_3,e_5\}$ 是一个粗糙集，下面分别计算集合 X 的上近似集、下近似集、正域、边界域。

首先计算论域 U 在属性集 B 下的划分：

$$U/\mathrm{IND}(B) = \{\{e_1,e_2,e_3\},\{e_4,e_6\},\{e_5\}\}$$

令 $B_1=\{e_1,e_2,e_3\}$，$B_2=\{e_4,e_6\}$，$B_3=\{e_5\}$，集合 X 与基本集有如下关系：

$$B_1 \not\subset X$$

$$B_2 \not\subset X$$

$$B_3 \subset X$$

$$X \cap B_1 = \{e_2,e_3\} \neq \varnothing$$

$$X \cap B_2 = \varnothing$$

$$X \cap B_3 = B_3 = \{e_5\} \neq \varnothing$$

由此可得集合 X 的上近似集、下近似集如下：

$$\overline{B}(X) = B_1 \cup B_3 = \{e_1,e_2,e_3,e_5\}$$

$$\underline{B}(X) = B_3 = \{e_5\}$$

图 10-1 所示给出下近似、上近似的直观描述。

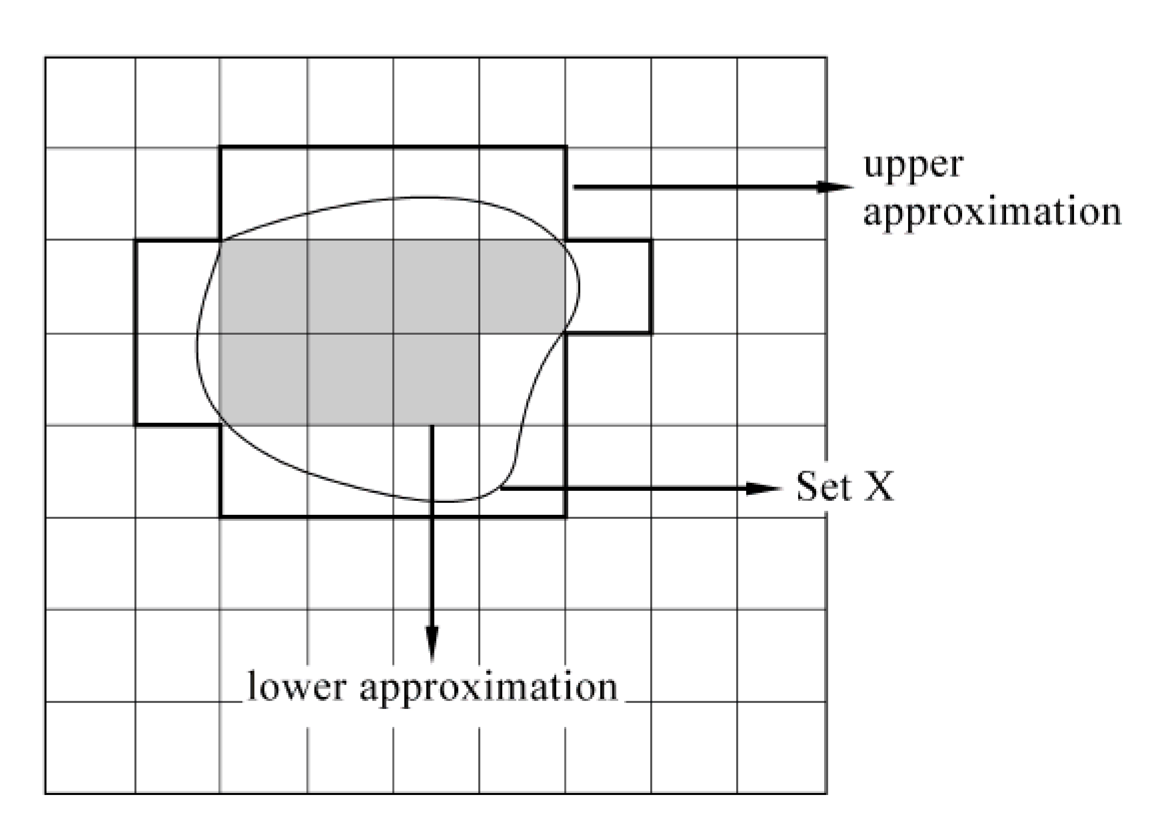


图 10-1　下近似、上近似的直观描述

10.2　基于粗糙集的属性约简

10.2.1　属性约简的有关概念

1. 正域

在信息表 S 中，对于属性集 $P,Q\subseteq A$，则 Q 的 P 正域 $\mathrm{POS}_P(Q)=\bigcap_{X\in U/Q}\underline{P}(X)$。

2. 属性约简

在决策表 L 中，若存在属性集 $R\subset C$，称 R 为相对于决策属性 D 的条件属性集 C 的约简，当且仅当满足两个条件：

(1) $\mathrm{POS}_R(D)=\mathrm{POS}_C(D)$；

(2) 不存在 $r\in R$，使得 $\mathrm{POS}_{R-\{r\}}(D)=\mathrm{POS}_C(D)$。

它表示相对约简后的部分条件属性形成的相对于决策属性的分类和所有条件属性形成的相对于决策属性的分类一致，即和所有条件属性相对于决策属性有相同的分类能力。

3. 最小属性约简

若决策表 L 的一个相对约简中所含条件属性个数为所有相对约简中最少的，则称该相对约简为决策表 L 的最小相对属性约简。

4. 属性核

在决策表 L 中，相对于决策属性集的条件属性的约简不一定是唯一的，设 $\mathrm{RED}_D(C)$ 为 C 的所有 D 约简簇，则称 $\bigcap\mathrm{RED}_D(C)$ 为 C 的 D 核，简称为决策表的相对属性核，记为 $\mathrm{CORE}_D(C)$。

例如，对表 10-3 所示的信息表，利用粗糙集可以进行如下属性约简。

令 Q=决策属性$\{d\}$，P=条件属性全集$\{a_1,a_2,a_3,a_4\}$，则

$U/\mathrm{IND}(P)=\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\},\{11\},\{12\},\{13\},\{14\}\}$

$U/\mathrm{IND}(Q)=\{\{1,2,6,8,14\},\{3,4,5,7,9,10,11,12,13\}\}$

$\mathrm{POS}_P(Q)=U$

表 10-3　一个信息表实例

U	条件属性				决策属性(d)
	Outlook(a_1)	Temperature(a_2)	Humidity(a_3)	Windy(a_4)	
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

因此,论域U是P上相对于Q一致的,这说明该决策表是完全确定的决策表,决策表中不包含不一致信息。

$$U/\mathrm{IND}(P\backslash\{a_1\}) = \{\{1,3\},\{2\},\{4,8\},\{5,9\},\{6,7\},\{10\},\{11\},\{12,14\},\{13\}\}$$

$$U/\mathrm{IND}(P\backslash\{a_2\}) = \{\{1,8\},\{2\},\{3\},\{4\},\{5,10\},\{6\},\{7\},\{9\},\{11\},\{12\},\{13\},\{14\}\}$$

$$U/\mathrm{IND}(P\backslash\{a_3\}) = \{\{1\},\{2\},\{3,13\},\{4,10\},\{5\},\{6\},\{7\},\{8\},\{9\},\{11\},\{12\},\{14\}\}$$

$$U/\mathrm{IND}(P\backslash\{a_4\}) = \{\{1,2\},\{3\},\{4,14\},\{5,6\},\{7\},\{8\},\{9\},\{10\},\{11\},\{12\},\{13\}\}$$

从而

$$\mathrm{POS}_{(P\backslash\{a_1\})}(Q) = \{2,5,9,10,11\}$$

$$\mathrm{POS}_{(P\backslash\{a_2\})}(Q) = U = \mathrm{POS}_P(Q)$$

$$\mathrm{POS}_{(P\backslash\{a_3\})}(Q) = U = \mathrm{POS}_P(Q)$$

$$\mathrm{POS}_{(P\backslash\{a_4\})}(Q) = \{1,2,3,7,8,9,10,11,12,13\}$$

由此可知,属性a_2,a_3是相对于决策属性d可省略的,而属性a_1和a_4是相对于决策属性d不可省略的,因此

$$\mathrm{CORE}_Q(P) = \{a_1,a_4\}$$

进一步,

$$U/\mathrm{IND}(P\backslash\{a_2,a_3\}) = \{\{1,8,9\},\{2,11\},\{3,13\},\{4,5,10\},\{6,14\},\{7,12\}\}$$

$$\mathrm{POS}_{(P\backslash\{a_2,a_3\})}(Q) = \{3,4,5,6,7,10,12,13,14\}$$

故属性a_2是条件属性集$P\backslash\{a_3\}$相对于决策属性d不可省略的,属性a_3也是条件属性$P\backslash\{a_2\}$相对于决策属性d不可省略的。条件属性集$\{a_1,a_3,a_4\}$和$\{a_1,a_2,a_4\}$为相对于决策属性集$Q=\{d\}$的条件属性P的约简,即

$$\mathrm{RED}_Q(P) = \{a_1,a_3,a_4\}$$

或

$$\mathrm{RED}_Q(P) = \{a_1,a_2,a_4\}$$

$$\mathrm{CORE}_Q(P) = \bigcap \mathrm{RED}_Q(P) = \{a_1, a_3, a_4\} \cap \{a_1, a_2, a_4\} = \{a_1, a_4\}$$

10.2.2 基于粗糙集的几种属性约简算法

1. 属性逐步删除约简算法

该算法的基本思想是对于决策表中的所有条件属性逐个进行删除，直至剩余的条件属性集合和原来的所有条件属性集合具有相同的分类能力。该算法一定能得到相对约简，但不一定能得到最小属性约简。

2. 属性逐步增加约简算法

该算法的基本思想是从属性核出发，逐步分别增加属性，直到和原来的所有条件属性具有相同的正域。该算法的实质是用幂子集的方法来求取最小相对约简，该方法的复杂性是指数性的，即当条件属性个数增加时，复杂性成指数增长。

3. 基于互信息的属性约简算法

该算法的基本思想同属性逐步增加约简算法相似，但它将互信息作为衡量属性重要程度的依据，一个一个增加属性，直到和原来的所有条件属性具有相同的正域。这种算法常常得不到最小约简，甚至根本得不到约简。

4. 基于可辨识矩阵和逻辑运算的属性约简算法

(1) 可辨识矩阵的概念。在决策表 $L=\langle U, C\cup D, V, f\rangle$ 中 $U=\{u_1, u_2, \cdots, u_n\}$，条件属性集合 $C=\{c_1, c_2, \cdots, c_m\}$，决策属性集合 $D=\{d\}$，设 E_{ij} 为 L 的可辨识矩阵的第 i 行第 j 列的元素，则可辨识矩阵 $\boldsymbol{E}$ 定义为

$$E_{ij} = \begin{cases} \{c_k \mid c_k \in C \text{ 且 } c_k(u_i) \neq c_k(u_j)\}, & d(u_i) \neq d(u_j) \\ 0, & d(u_i) = d(u_j) \end{cases} \tag{10-6}$$

其中 $i, j = 1, \cdots, n$。

对于表 10-3 所示的决策表，可以得到如下所示的可辨识矩阵。

$$\begin{bmatrix}
0 & 0 & a_1 & a_1a_2 & a_1a_2a_3 & 0 & a_1a_2a_3a_4 & 0 & a_2a_3 & a_1a_2a_3 & a_2a_3a_4 & a_1a_2a_4 & a_1a_3 & 0 \\
 & 0 & a_1a_4 & a_1a_2a_4 & a_1a_2a_3a_4 & 0 & a_1a_2a_3 & 0 & a_2a_3a_4 & a_1a_2a_3a_4 & a_2a_3 & a_1a_2 & a_1a_3a_4 & 0 \\
 & & 0 & 0 & 0 & a_1a_2a_3a_4 & 0 & a_1a_2 & 0 & 0 & 0 & 0 & 0 & a_1a_2a_4 \\
 & & & 0 & 0 & a_2a_3a_4 & 0 & a_1a_2 & 0 & 0 & 0 & 0 & 0 & a_4 \\
 & & & & 0 & a_4 & 0 & a_1a_3 & 0 & 0 & 0 & 0 & 0 & a_2a_3a_4 \\
 & & & & & 0 & a_1 & 0 & a_1a_4 & a_2a_4 & a_1a_2 & a_1a_2a_3 & a_1a_2a_4 & 0 \\
 & & & & & & 0 & a_1a_2a_3a_4 & 0 & 0 & 0 & 0 & 0 & a_1a_2a_3 \\
 & & & & & & & 0 & a_2a_3 & a_1a_3 & a_3a_4 & a_1a_4 & a_1a_2a_3 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 0 & 0 & a_1a_2a_3a_4 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & a_3a_4 \\
 & & & & & & & & & & 0 & 0 & 0 & a_1a_3 \\
 & & & & & & & & & & & 0 & 0 & a_1a_4 \\
 & & & & & & & & & & & & 0 & a_1a_2a_3a_4 \\
 & & & & & & & & & & & & & 0
\end{bmatrix}$$

显然，可辨识矩阵是一个以主对角线对称的矩阵，在考虑可辨识矩阵的时候，只要考虑其上三角(或下三角)部分即可。如果在可辨识矩阵中存在某元素，其取值为单属性集合，则

表明该属性是区分这个矩阵元素所对应的两个对象所必需的属性，也是唯一能够区分这两个对象的属性。可辨识矩阵中的这些元素所包含的属性组成的属性集合其实就是该决策表的相对属性核。

（2）基于可辨识矩阵和逻辑运算的属性约简算法。

第 1 步，计算决策表的可辨识矩阵。

第 2 步，对于可辨识矩阵中的所有取值为非空集合的元素，建立相应的析取逻辑表达式。

第 3 步，将所有的析取逻辑表达式进行合取运算，得到一个合取范式。

第 4 步，将合取范式转换为析取范式的形式。

第 5 步，输出属性约简结果。析取范式中的每个合取项就对应一个属性约简的结果，每个合取项中所包含的属性组成约简后的条件属性集合。

对于表 10-3 所示的决策表，我们利用以上算法进行属性约简的过程如下。

$$L_{1,3}=a_1$$
$$L_{2,3}=a_1 \vee a_4$$
$$L_{1,4}=a_1 \vee a_2$$
$$L_{2,4}=a_1 \vee a_2 \vee a_4$$
$$\vdots$$
$$L_{13,14}=a_1 \vee a_2 \vee a_3 \vee a_4$$

将这些表达式进行合取得到合取表达式 L

$$\begin{aligned} L &= L_{1,3} \wedge L_{2,3} \wedge L_{1,4} \wedge L_{2,4} \wedge \cdots \wedge L_{13,14} \\ &= a_1 \wedge (a_1 \vee a_4) \wedge (a_1 \vee a_2) \wedge (a_1 \vee a_2 \vee a_4) \wedge \cdots \wedge (a_1 \vee a_2 \vee a_3 \vee a_4) \end{aligned}$$

对 L 进行转换，最终得到析取范式 L'

$$L'=(a_1 \wedge a_2 \wedge a_4) \vee (a_1 \wedge a_3 \wedge a_4)$$

这样就得到了两个属性约简结果，分别为 a_1,a_2,a_4 和 a_1,a_3,a_4。

基于可辨识矩阵和逻辑运算的属性约简算法可以得到所有可能属性约简结果，但算法的实质是将属性组合情况的搜索演变为逻辑公式的化简，没有解决 NP 问题。

10.3　基于粗糙集的决策规则约简

10.3.1　决策规则的定义

信息表中的对象（元组）x 按条件属性与决策属性关系看成一条决策规则，写成

$$\wedge f_{c_i}(x) \rightarrow f_d(x)$$

其中 C_i 表示多个条件属性，d 表示决策属性，$f_{c_i}(x)$表示对象 x 在属性 C_i 的取值，$\wedge$ 表示逻辑“与”关系。

对表 10-4 信息表的决策规则如下：

① $a_1 b_0 c_2 \rightarrow d_1 e_1$

② $a_2 b_1 c_0 \rightarrow d_2 e_0$

③ $a_2 b_1 c_2 \rightarrow d_0 e_2$

④ $a_1 b_2 c_2 \rightarrow d_1 e_1$

⑤ $a_1 b_2 c_0 \rightarrow d_0 e_2$

注：a_i 即 $a=i(i=1,2)$；b_j 即 $b=j(j=1,2)$；其他类同。

表 10-4　信息表实例

U	a	b	c	d	e
1	1	0	2	1	1
2	2	1	0	2	0
3	2	1	2	0	2
4	1	2	2	1	1
5	1	2	0	0	2

10.3.2　决策规则的约简

利用粗糙集理论可以得到以下决策规则约简命题：

若属性集 C 中存在若干个属性，关于这些属性的等价类的交集为相应决策属性 D 的等价类的交集的子集，即

$$\bigcap [x]_a \subseteq \bigcap [y]_d$$

则由这些条件属性和全部决策属性构成的新决策规则是原决策规则的约简。

参照表 10-4，求每一条决策规则的约简。

第一条规则的约简如下：

$$[1]_{\{d,e\}} = \{1,4\}; \quad [1]_a = \{1,4,5\}; \quad [2]_c = \{1,3,4\}$$

显然$[1]_a \not\subset [1]_{\{d,e\}}$；$[2]_c \not\subset [1]_{\{d,e\}}$，$[0]_b = \{1\} \subseteq [1]_{\{d,e\}}$；$[1]_a \cap [1]_c = \{1,4\} \subseteq [1]_{\{d,e\}}$，所以得到两条约简后的决策规则：

$$1: b_0 \rightarrow d_1 e_1, 1': a_1 c_2 \rightarrow d_1 e_1$$

第二条规则的约简如下：

$$[2]_d \cap [0]_e = \{2\}; \quad [2]_a = \{2,3\}; \quad [1]_b = \{2,3\}; \quad [0]_c = \{2,5\}$$

显然有：

$$[2]_a \cap [0]_c = \{2\} \subseteq [2]_d \cap [0]_e, \quad [1]_b \cap [0]_c = \{2\} \subseteq [2]_d \cap [0]_e$$

所以得到两条约简规则如下：

$$2: a_2 c_0 \rightarrow d_2 e_0; 2': b_1 c_0 \rightarrow d_2 e_0$$

同样可得，3、4、5 条规则的约简，它们分别为：

$$3: a_2 c_2 \rightarrow d_0 e_2; 3': b_1 c_2 \rightarrow d_0 e_2$$

$$4: a_1 c_2 \rightarrow d_1 e_1; 4': b_2 c_2 \rightarrow d_1 e_1$$

$$5: a_1 c_0 \rightarrow d_0 e_2; 5': b_2 c_0 \rightarrow d_0 e_2$$

所有约简的决策规则如表 10-5 所示。

表 10-5　约简后的决策规则

U	a	b	c	d	e
1	*	0	*	1	1
1′	1	*	2	1	1
2	2	*	0	2	0

续表

U	a	b	c	d	e
2′	*	1	0	2	0
3	2	*	2	0	2
3′	*	1	2	0	2
4	1	*	2	1	1
4′	*	2	2	1	1
5	1	*	0	0	2
5′	*	2	0	0	2

注：1′和 4 规则相同，可以合并。

10.4　粗糙集的优缺点

10.4.1　粗糙集的优点

(1) 粗糙集不需要先验知识。模糊集和概率统计方法是处理不确定信息的常用方法，但这些方法需要一些数据的附加信息或先验知识，如模糊隶属函数和概率分布等，这些信息有时并不容易得到。粗糙集分析方法仅利用数据本身提供的信息，无须任何先验知识。

(2) 粗糙集是一个强大的数据分析工具。它能表达和处理不完备信息；能在保留关键信息的前提下对数据进行化简并求得知识的最小表达；能识别并评估数据之间的依赖关系，揭示出概念简单的模式；能从经验数据中获取易于证实的规则知识，特别适于智能控制。

10.4.2　粗糙集的缺点

粗糙集理论的缺点是在实际应用中由于对象间等价关系(不可分辨关系)条件过分严格，以及数据中存在噪声和数据不完整，会造成知识的遗漏和偏差。如以下例子：$E1$，$E2$ 是条件属性下的两个等价类，$E1$，$E2$ 分别有 100 个元素(对象)，对某一决策属性等价类 X，$E1$ 中只有一个元素属于 X，而 $E2$ 只有一个元素不属于 X，基于一般的粗糙集模式，$E1$，$E2$ 同属于 X 的边界区，对 X 都不能推出肯定的判断，然而 $E2$ 中仅有一个元素不属于 X，也许是由噪音导致的；这也说明一般粗糙集对噪声是非常敏感的。而在现实的应用中，获取的数据含有噪声是不可避免的，因此一般粗糙集模式在一定程度上限制了其更有效的应用。

小结

本章首先介绍了粗糙集中不可分辨关系、上近似和下近似等基本概念。然后在引入正域概念的基础上介绍了基于粗糙集的属性约简算法，其中主要包括基于互信息的属性方法、基于可辨识矩阵和逻辑运算的属性约简方法。接下来通过实例介绍了基于粗糙集的决策规则约简方法。最后介绍了粗糙集方法的优缺点。

习题 10

1. 信息表和决策表有什么异同?
2. 解释上近似和下近似的概念。
3. 解释属性约简的概念。
4. 基于粗糙集的属性约简算法有哪几种,并解释它们的约简思路。
5. 阐述如何利用粗糙集进行决策规则的约简。
6. 粗糙集理论具有哪些优点和缺点?

第 11 章　复杂结构数据挖掘

知识不仅以传统数据库的结构化数据形式出现，它还以各种各样的半结构化或非结构化数据形式存储、表现，诸如文本数据、空间数据、多媒体数据、Web 页面等。复杂结构的数据源中存在着大量的知识，从这些数据源中进行数据挖掘，提取知识就成为数据挖掘的一个发展方向。

11.1　文本数据挖掘

在现实世界，人们面对大量的文本数据，由各种数据源（如新闻文章、研究论文、书籍、数字图书馆、电子邮件和 Web 页面）的大量文本组成。随着信息技术的发展，文本数据的数量急剧增长，所以对文本进行数据挖掘成为了数据挖掘的一个发展方向。

11.1.1　文本数据的特点

文本数据不同于传统数据库中的数据，它具有自己的特点。

（1）半结构化。文本数据既不是完全无结构的也不是完全结构化的。例如文本可能包含结构字段，如标题、作者、出版日期、长度、分类等，也可能包含大量的非结构化的数据，如摘要和内容。

（2）高维。文本向量的维数一般都可以高达上万维，一般的数据挖掘、数据检索的方法由于计算量过大或代价高昂而不具有可行性。

（3）高数据量。一般的文本库中都会存在最少数千个文本样本，对这些文本进行预处理、编码、挖掘等处理的工作量是非常庞大的，因而手工方法一般是不可行的。

（4）语义性。文本数据中存在着一词多义、多词一义，在时间和空间上的上下文相关等情况。

11.1.2　文本挖掘的定义

文本挖掘以文本型信息源作为分析的对象，利用定量计算和定性分析的方法，从中寻找到信息结构、模型、模式等各种隐含的新颖知识。

文本挖掘与数据挖掘相比，它们的相似点在于两者都处理大量的数据，都可归属到知识发现领域中；其差别在于许多经典的数据挖掘算法，如数值预测、决策树等都不太适用于文本挖掘，因为数据挖掘依赖于结构化的数据；另外短语或概念关联分析等工作则是文本挖掘所独有的。

文本挖掘和数据挖掘之间的区别如表 11-1 所示。

表 11-1　文本挖掘与数据挖掘的区别

项目	数据挖掘	文本挖掘
研究对象	用数字表示的、结构化的数据	半结构化的数据
对象结构	关系数据库	自由开放的文本
目标	获取知识、预测以后的状态	提取概念和知识
方法	归纳学习、决策树、神经网络、粗糙集等	提取短语、形成概念、关联分析、聚类、分类
成熟度	1994 年始得到广泛应用	2000 年始得到广泛应用

11.1.3　文本挖掘的主要任务

（1）文本预处理。文本中包含的内容非常复杂，为了降低文本挖掘的计算量和复杂度、排除噪声，一般必须对文本进行预处理。常见的预处理步骤为：分词和预过滤，以避免非常短或者非常长的关键词以及不是单词的关键词；高、低通过滤器，过滤那些很不常用和诸如辅助动词那些出现频率很高的常用词；词频统计，以词频为自变量的函数是目前流行的各类以统计方法为数学基础的文本处理方法的基本处理对象。

（2）文本结构分析和自动摘要。进行文本结构分析是为了更好地理解文本的主题思想，了解文本所表达的内容及采用的方式，包括识别文本的各个层次及分析各个层次间的联系。以结构分析为基础，找出文本主题句，整理组合后构成文本文摘。

（3）文本分类。文本分类是指按照预先定义的主题类别，为文本集合中的每个文本确定一个类别。这样用户不但能够方便地浏览文本，而且可以通过限制搜索范围来使文本的查找更容易、快捷。

（4）文本聚类。聚类与分类的不同之处在于，聚类没有预先定义好的主题类别，它的目标是将文本集合分成若干个簇，要求同一簇内文本内容的相似度尽可能的大，而不同簇之间的相似度尽可能的小。

11.1.4　文本挖掘的一般过程

文本挖掘过程一般包括文本分词、文本特征表示、词频矩阵降维、文本相似度计算、文本知识获取等。

在经过对文本数据进行一系列的预处理以后，传统的数据挖掘方法同样可以应用于文本数据挖掘，下面简要介绍文本的预处理方法。

1. 文本分词

分词是中文信息处理从字符处理水平向语义处理水平迈进的关键。汉语文本不像西文那样，词与词之间有空格间隔，同时汉语的构词方式、不同分词方式表达不同意义等特点，使得中文处理必须有分词这道工序。

汉语分词的难点主要表现在两个方面，即歧义切分和未登录词的切分。

（1）歧义切分。汉语字与字之间组词灵活，给分词带来了很大的困难。从上下文关系的角度看，其中只有一种切分结果是正确的。

（2）未登录词切分。未登录词主要是指分词系统的词典中未收录的词。不断出现的新词属于另外一类未登录词，反映在自然语言上就是大量的新词不断涌现。

分词技术，大致可以分为 5 类：词典分词法、切分标记分词法、基于统计的分词方法、基

于语言规则的分词方法和智能分词方法。

（1）词典分词法。词典分词法主要用于主题相对集中的信息库，如某专业学科信息库。

就扫描的顺序而言，词语匹配方法有正向扫描匹配、逆向扫描匹配和正逆向结合扫描匹配；在进行词语匹配时，有最长匹配、最短匹配、长短匹配结合、词首匹配等多种策略。例如最短匹配是指以短字符串优先匹配词典。

（2）切分标记分词法。利用切分字典指导分词。切分字典是由能断开词和词组或表示汉字之间关系的汉字集合组成字典，包括的内容有词首字、词尾字等，也有的系统以非用字、条件用字等组成切分字典。

（3）基于统计的分词方法。用字与字相邻共现的频率来反映字符串确实是一个词的可信程度。在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。

（4）基于语言规则的分词方法。在分词的过程中加入词法、语法以及语义规则等来提高分词的质量。一般都是人工添加规则，或者在人工添加的基础上再从有限的训练语料库中得到分词规则。

（5）智能分词法。利用人工智能的方法进行分词。常用的有中心词驱动分析法、分词语句法、语义分析同步处理法和分层理解分析法等。

2. 文本特征表示

文本特征指的是关于文本的元数据，分为两种：描述性特征，例如文本的名称、日期、大小、类型等；语义性特征，例如文本的作者、机构、标题、内容等。描述性特征易于获得，而语义性特征则较难得到。

向量空间模型是近年来应用较多且效果较好的表示文本特征的方法。在该模型中，文本空间被看做是由一组正交词条向量所张成的空间，每一个词条称为一个特征项，每一个文本 d 则表示为空间内的一个向量，一般表示为：

$$V(d) = (w(t_1), w(t_2), w(t_3), \cdots, w(t_n)) \tag{11-1}$$

其中 t_i 为张成文本空间的词条，n 为文本空间的维数，$w(t_i)$是函数，其基本功能是计算词条 t_i 在文本向量中的权重。$w(t_i)$一般被定义为 t_i 在文本 d 中出现频率 $tf_i(d)$的函数，即 $w(t_i)=\psi(tf_i(d))$，常用的 ψ 有以下几种函数。

（1）布尔函数：

$$\psi = \begin{cases} 1, & tf_i(d) > 0 \\ 0, & tf_i(d) = 0 \end{cases} \tag{11-2}$$

（2）平方根函数：

$$\psi = \sqrt{tf_i(d)} \tag{11-3}$$

（3）对数函数：

$$\psi = \log(tf_i(d) + 1) \tag{11-4}$$

（4）TFIDF 函数：

$$\psi = tf_i(d) \times \log\left(\frac{N}{n_i}\right) \tag{11-5}$$

其中 N 为所有文本的数目，n_i 为含有词条 t_i 的文本数目。

3. 词频矩阵降维

使用向量空间模型来表示文本时，表示文本的特征向量会达到数十万维的大小。有人

曾对 Yahoo 上的 49 600 个文本提取作为特征的词串，最后得到 320 000 个特征词串。如此高维的特征对将进行的机器学习未必全是重要的、有益的，而且高维的特征可能会大大增加数据挖掘时间而仅产生与小得多的特征子集相关的学习结果，因此对文本进行数据挖掘时特征子集的选取便显得异常重要。下面介绍两种常用的词频矩阵降维方法。

（1）基于评估函数的方法。基于评估函数的特征集缩减算法一般使用特征独立性假设以简化特征选择。算法的一般步骤是，用某种评估函数独立地对每个特征计算，然后把特征按计算结果进行排序，选择事先确定数目或者超过阈值的若干计算值较高的特征。

选择不同的评估函数会形成各种名目的特征集缩减算法。常见的评估函数有信息增益、期望交叉熵、互信息、文本频数、文本证据权、优势率、词频等。

设 F 是对应于单词 W 的特征，$P(W)$ 是单词 W 出现的概率，$P(c_i)$ 为第 i 类出现的概率，$P(C_i|W)$ 为单词 W 出现时属于第 i 类的概率，$TF(W)$ 为单词 W 在文本集中出现的次数，pos 表示正例集的情况，neg 表示负例集的情况。以上评估函数分别定义如下：

文本频数：$TF(W)$

信息增益：

$$P(W)\sum_i P(C_i \mid W)\log\left(\frac{P(C_i \mid W)}{P(C_i)}\right)+P(\overline{W})\sum_i P(C_i \mid \overline{W})\log\left(\frac{P(C_i \mid \overline{W})}{P(C_i)}\right) \tag{11-6}$$

期望交叉熵：

$$P(W)\sum_i P(C_i \mid W)\log\left(\frac{P(C_i \mid W)}{P(C_i)}\right) \tag{11-7}$$

互信息：

$$\sum_i P(C_i)\log\left(\frac{P(WC_i)}{P(W)}\right) \tag{11-8}$$

文本证据权：

$$P(W)\sum_i P(C_i)\left|\log\left(\frac{P(C_i \mid W)(1-P(C_i))}{P(C_i)(1-P(C_i \mid W))}\right)\right. \tag{11-9}$$

优势率：

$$\log\left(\frac{P(W \mid \mathrm{pos})(1-P(W \mid \mathrm{neg}))}{P(W \mid \mathrm{neg})(1-P(W \mid \mathrm{pos}))}\right) \tag{11-10}$$

特征的 χ^2 统计：单词 W 在类别 C_i 中的 χ^2 值计算如下：

$$\chi_{ji}^2=\frac{n(n_1\times n_4-n_2\times n_3)^2}{(n_1+n_2)\times(n_3+n_4)\times(n_1+n_3)\times(n_2+n_4)} \tag{11-11}$$

其中 n_1、n_2、n_3 和 n_4 分别表示单词 W 和类别 C_i 的 4 种情况的频数，这 4 种情况分别为 (W,C_i)；$(W,\overline{C}_i)$；$(\overline{W},C_i)$ 和 $(\overline{W},\overline{C}_i)$；且 $n=n_1+n_2+n_3+n_4$ 为总数。

（2）潜在语义索引（latent semantic index，LSI）。对文本词条矩阵 $\boldsymbol{W}_{N\times M}$ 利用奇异值分解计算 $\boldsymbol{W}$ 的 r-秩近似矩阵 $\boldsymbol{W}_r$（$r\ll\min(N,M)$）。LSI 分解图示如图 11-1 所示。

经奇异值分解，矩阵 $\boldsymbol{W}$ 可表示为三个矩阵的乘积：$\boldsymbol{W}=\boldsymbol{UAV}^{\mathrm{T}}$，$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 分别为与 $\boldsymbol{W}$ 矩阵对应的左、右奇异向量矩阵，$\boldsymbol{A}$ 为由矩阵 $\boldsymbol{W}$ 的奇异值按递减顺序排列构成的对角矩阵。取 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 最前面的列构建 r-秩近似矩阵 $\boldsymbol{W}_r$：

$$\boldsymbol{W}_r=\boldsymbol{U}_r\boldsymbol{A}_r\boldsymbol{V}_r^{\mathrm{T}} \tag{11-12}$$

图 11-1　LSI 分解图示

用 $\boldsymbol{W}_r$ 近似表示文本词条矩阵 $\boldsymbol{W}$,$\boldsymbol{U}_r$ 和 $\boldsymbol{V}_r$ 行向量分别作为文本向量和词向量。在此基础上进行文本聚类或其他处理。

4. 文本相似度计算

一般认为相似的文本具有相似的关键词和相对词频,一组文本的相似度可以基于关键词向量或相对词频向量来计算,利用文本的相似度可以对文本进行分类和聚类。

在文本与词的相关矩阵($t_{ij}/i=1,2,\cdots,m;j=1,2,\cdots,n$)中,$t_{ij}$ 表示第 i 篇文本与第 j 个词的相关度,t_{ij} 的取值范围为[0,1]。

第 i 篇文本与第 j 篇文本的相关系数用 S_{ij} 表示。计算相关系数的方法有多种,其中,余弦系数法最为常用。

$$S_{ij}=\frac{\sum_{k=1}^{n}(t_{ik},t_{jk})}{\sqrt{\sum_{k=1}^{n}t_{ik}^{2}}\sqrt{\sum_{k=1}^{n}t_{jk}^{2}}} \tag{11-13}$$

设 d_1 和 d_2 为两个文本特征向量,它们的余弦相似度定义也可以表示成如下形式:

$$\cos(d_1,d_2)=(d_1\cdot d_2)/|d_1||d_2| \tag{11-14}$$

其中 $d_1\cdot d_2$ 为标准向量乘积,分母中的 $|d_1|$ 是向量 d_1 的长度,$|d_2|$ 是向量 d_2 的长度。

11.1.5 文本挖掘的应用

电子邮件管理是文本挖掘应用的成功案例。利用文本挖掘构造的电子邮件路由,可以对电子邮件进行文本挖掘以后,确定由哪一个部门、哪一个人来处理这些电子邮件,并且可以根据电子邮件的内容进行相关统计。

此外,文档管理是许多组织中十分烦琐而又重要的工作,通过文本挖掘可以帮助组织对成千上万的文档实现有效的管理,很快地了解到所需要查找文档的所在位置以及其包含的主要内容。

企业可以利用文本挖掘建立一个客户自动问答系统。对客户所邮寄的信件、电子邮件进行文本挖掘后,根据其反映的主要问题,在确定了客户的需求置信度后,可以自动给客户发送合适的回信。

企业还可以利用联机文本挖掘系统对因特网上所出现的特定词、概念、主题进行挖掘统计,对市场进行客观的统计分析。

企业甚至可以利用一些具有文本挖掘功能的自动智能网络爬虫来收集与企业有关的市场、竞争对手和市场环境的信息,给出总结性的分析报告。

11.2 Web 数据挖掘

随着 Internet 技术的发展,尤其是 Web 的全球普及,使得 Web 成为信息发布、交互和获取的主要工具。Web 上的数据是海量的,同时 Web 是半结构的、动态的,Web 页面极其复杂,这样就使人们从成千上万的 Web 站点中找到所需数据极为困难。而 Web 挖掘就可以利用数据挖掘技术从 Web 文档和 Web 活动中抽取人们感兴趣的、潜在的有用模式和隐藏的信息。

11.2.1 Web 数据的特点

Web 是一个非常成功的基于超文本的分布式信息系统，它包括了丰富的动态的超链接信息，以及 Web 页面的访问和使用信息，为数据挖掘提供了丰富的资源。要了解 Web 挖掘，首先要了解其特点。

(1) 庞大性。Web 为全球范围发布和传播信息提供了机会，它允许任何人在任何地点任何时间传播和获取信息。由于 Web 的开放性，使得其数据与日俱增，爆炸性增长。

(2) 动态性。Web 不仅以极快的速度增长，而且其信息还在不断地发生更新。新闻、公司动态、股票市场、Web 服务中心等都在不断地更新着各自的页面。链接信息和访问记录也在频繁更新之中。

(3) 异构性。从数据库研究的角度出发，Web 网站上的信息可以看作是一个庞大、复杂的数据库。Web 上的每一个站点就是一个数据源，每个数据源都是异构的，这就是构成了一个巨大的异构数据库环境。

(4) 半结构化数据结构。Web 上的数据与传统数据库中的数据不同，传统数据库都有一定的数据模型，Web 上的数据非常复杂，没有特定的模式描述，每一个站点的数据都是各自独立设计，并且数据本身具有自述性和动态可变性。因而，Web 上的数据应具有一定的结构性，但因其自述层次的存在，从而是一种非完全结构化的数据，即半结构化数据。半结构化是 Web 数据的最大特点。

11.2.2 Web 挖掘的定义

Web 挖掘可以简单地定义为，针对包含 Web 页面内容、页面之间的结构、用户访问信息、电子商务信息等在内的各种 Web 数据，应用数据挖掘方法以帮助人们从 WWW 中提取知识，为访问者、站点经营者以及包括电子商务在内的基于因特网的商务活动提供决策支持。

11.2.3 Web 挖掘分类

Web 数据有 3 种类型：HTML 标记的 Web 文档数据，Web 文档内连接的结构数据和用户访问数据。按照对应的数据类型，Web 挖掘可分为内容挖掘、结构挖掘、用户访问模式挖掘。

1. Web 内容挖掘

Web 内容挖掘主要是针对各种非结构化数据(如文本数据、音频数据、视频数据和图形图像数据等各种数据相融合的多媒体数据)的挖掘。可以将其分为基于文本信息的挖掘和基于多媒体信息的挖掘两种数据挖掘方式。

(1) 基于文本信息的挖掘。Web 上的内容挖掘多为基于文本信息的挖掘，它和通常的平面文本挖掘比较类似。平面文本挖掘的方法也可以用到 Web 文本的挖掘。Web 文档多为 HTML、XML 等语言，因此可以利用 Web 文档中的标记额外信息来提高 Web 文本挖掘的功能。

在对 Web 文档进行分类分析中，可以基于一组预先分好类的文档，从预定义好的分类目录中为每一文档赋予一个类标签，用于导出 Web 文档的分类模式。接下来可以利用导出

的分类模式对 Web 文档进行分类。而且,由于超链接包含了有关页面的高质量信息,因此,可以利用这些信息对 Web 文档进行分类。这种分类比基于关键字的分类方法更准确、更完善。

(2) 基于多媒体信息的挖掘。随着网络带宽的不断增加,多媒体信息在网上迅速增加,这就对多媒体信息的挖掘提出了要求。多媒体信息的挖掘主要是指基于音频的挖掘、基于静态图像的挖掘和基于动态图像的挖掘。

2. Web 结构挖掘

Web 结构挖掘是从 WWW 的组织结构和链接关系中推导知识。Web 结构挖掘通过分析一个网页链接和被链接的数量及对象建立 Web 自身的链接结构模式。这种模式可以用于网页归类,并且可以由此获得有关不同页面相似度和关联度的信息,Web 结构挖掘有助于用户找到相关主题的权威站点,并且可以指向众多权威站点的相关主题站点。

3. 用户访问模式挖掘

Web 内容挖掘和 Web 结构挖掘的挖掘对象是网上的原始数据,而用户访问模式挖掘则不同于前两者,它面对的是在用户和网络交互的过程中抽取出来的第二手数据。Web 用户访问模式挖掘时对用户访问 Web 时在服务器方留下的访问记录进行挖掘,即对用户访问 Web 站点的存取方式进行挖掘。挖掘的对象是在服务器上的包括 Server Log Data 等在内的日志文件记录。

11.2.4 Web 挖掘过程

(1) 查找资源。根据挖掘目的,从 Web 资源中提取相关数据,构成目标数据集。其任务是从 Web 数据(包括 Web 文档、电子邮件、电子文档、新闻组、网站日志、网络数据库中的数据等)中得到数据。

(2) 数据预处理。在进行 Web 挖掘之前对“杂质”数据进行过滤。例如消除数据的不一致性;将多个数据源中的数据统一为一个数据存储等。预处理数据的效果直接影响到挖掘算法产生的规则和模式。数据预处理主要包括站点识别、数据选择、数据净化、用户识别和会话识别等。

(3) 模式发现。利用挖掘算法挖掘出有效的、新颖的、潜在的、有用的及最终可以理解的信息和知识。常用的模式发现技术包括路径分析、关联规则挖掘、时序模式发现、聚类和分类等技术。

(4) 模式分析。利用合适的工具和技术对挖掘出来的模式进行分析、解释、可视化,把发现的规则模式转换为知识。

11.2.5 Web 数据挖掘的应用

Web 数据挖掘的应用涉及电子商务、网站设计和搜索引擎服务等多个方面。

在电子商务的应用方面主要有客户分类和客户聚类、寻找潜在的客户、客户的驻留。

网站设计的应用是通过对网站内容的挖掘,有效地组织网站信息,并结合对用户访问日志记录信息的挖掘,把握用户的兴趣,从而有助于开展网站信息服务以及个人信息的定制服务。

在搜索引擎服务方面主要通过对网页内容的挖掘,实现对网页的分类、聚类,实现网络

信息的分类浏览与检索;通过用户所使用的查询历史记录分析,提高用户的检索效果(查全率、查准率)。

由于 Web 上存在大量信息,并且 Web 在当今社会生活中扮演越来越重要的角色,发挥越来越大的作用,因此 Web 挖掘的应用将越来越广泛,用户对高品质、个性化信息的需求也将进一步推动 Web 挖掘这项技术的研究和开发。

11.3 空间数据挖掘

通过遥感、地理信息系统、医学和卫星图像等多种数据采集设备收集到了大量的空间数据,这些空间数据远远超过了人脑的分析能力。空间数据不同于关系数据,它一般具有空间拓扑或距离信息,通常需要以复杂的多维空间索引结构组织,另外空间数据的处理还常常需要空间推理、几何计算和空间知识表示技术。这些特性使得从空间数据中挖掘信息具有很多挑战性。

11.3.1 空间数据的复杂性特征

空间数据的复杂性特征主要表现在以下几个方面。

1. 空间属性之间的非线性关系

空间属性之间的非线性关系是空间系统复杂性的重要标志,其中蕴含着系统内部运作的复杂机制,因而被作为空间数据挖掘的主要任务之一。

2. 空间数据的多尺度特征

空间数据的多尺度性是指空间数据在不同观察层次上所遵循的规律以及体现出不尽相同的特征。多尺度特征是空间数据复杂性的又一表现形式,利用该性质可以探究空间信息在泛化和细化过程中所反映出的特征渐变规律。

3. 空间信息的模糊性

空间数据复杂性的另一个特征就是模糊性。模糊性几乎存在于各种类型的空间信息中,如空间位置的模糊性、空间相关性的模糊性以及模糊的属性值等。

4. 空间维数的增高

空间数据的属性增加极为迅速,如在遥感领域,由于传感技术的飞速发展,波段的数目也由几个增加到几十甚至上百个,如何从几十甚至几百维空间中提取信息、发现知识成为研究中的又一难点。

5. 空间数据的缺值

数据的缺值现象源自于某种不可抗拒的外力而使数据无法获得或发生丢失。如何对丢失数据进行恢复并估计数据的固有分布参数,成为解决数据复杂性的难点。

11.3.2 空间数据挖掘的定义

空间数据挖掘是指在空间数据库的基础上,综合利用各种技术方法,从大量的空间数据中自动挖掘事先未知的且潜在有用的知识,提取非显式存在的空间关系或其他有意义的模式等,揭示出蕴含在数据背后的客观世界的本质规律、内在联系和发展趋势,实现知识的自动获取,从而提供技术决策与经营决策的依据。它可以用来理解或重组空间数据、发现空间

和非空间数据间的关系、构建空间知识库、优化查询等。

11.3.3 空间数据挖掘知识的类型

空间数据挖掘知识的类型主要包括以下几种。

一般几何知识。目标的数量、大小、特征的统计特征值及直方图等可视化描述。

空间分布规律。垂直向、水平向及其联合向的分布规律。

空间关联规则。空间相邻、相连、共生、包含等空间关联规则。

另外还包括空间聚类规则、空间特征规则、空间区分规则、空间演变规则、空间序贯模式、空间混沌模式。

11.3.4 空间数据挖掘的用途

空间数据挖掘在生产生活实践中有着众多应用，下面通过举例简单说明空间数据分类、空间数据关联分析、空间趋势分析等的应用。

(1) 空间数据分类。企业在确定销售地点分布时，往往需要根据人口的家庭收入、接受教育水平等因素对全国、全球进行地区分类。此时，就需要找出决定地区分类的空间因素。例如大中学校、高速公路、服务设施等特性。通过对这些特性的分析找出有意义的分类模式，为企业寻找新的销售点提供合适的模式。

(2) 空间数据关联分析。同普通的关联规则挖掘一样，空间数据也可以进行关联规则挖掘。例如气象数据仓库就是一种典型的空间数据仓库，人们可以利用气象数据仓库中的数据挖掘与气象有关的生产活动规则。例如，农业生产、交通运输等生产活动与气象的关联规则。

(3) 空间趋势的分析。在分析一个地区经济发展与周边空间环境的变化趋势时，需要用到空间数据趋势挖掘。例如，在进行地区经济发展规划时，希望能够寻找到合适的模式：地区经济发展的不同模式与大中城市的联系、与交通的关联等。这种空间趋势分析，一般需要在空间数据结构和空间数据访问的基础之上，使用回归或者相关分析方法进行。

11.4 多媒体数据挖掘

11.4.1 多媒体数据挖掘的概念

多媒体数据挖掘指从大量的图像、视频、音频等多媒体数据集中，通过分析视听特征语义，发现隐含的、有效的有价值的、可以理解的模式，为用户提供问题层次的决策支持能力。

多媒体数据挖掘与计算机视角图像处理的区别是，前者的焦点是从多媒体中抽取一定的模式，后者的焦点是从单个图像中分析和提取特定的特征。

传统的数据挖掘与多媒体数据挖掘的区别是，前者处理的是关系数据库的结构化数据，后者处理的是非结构化的多媒体数据，多媒体的时间、空间、视听对象、运动特征是多维的，模式的表示是建立在丰富的视觉环境下。

11.4.2 多媒体挖掘的分类

按处理的媒体数据进行分类，可分为以下 3 种。

(1) 图像数据挖掘。从图像的视觉和空间特征中抽取有意义的语义信息,即知识。其根本的问题在于将低层特征如何关联转换为高层对象和语义概念。

(2) 视频数据挖掘。从含有图像视觉和空间特性、时间特性、视频对象特性、运动特性等的内容获取有意义的知识。如从交通监视视频中分析出交通拥塞的趋势。

(3) 音频数据挖掘。从听觉特性中的基音、音调、旋律、音频事件和对象的结构中挖掘出隐含在音频流中的信息线索、规律和特性。

小结

在现实生活中,存在着诸如文本数据、空间数据、多媒体数据等形式的复杂结构数据,如何对这些复杂结构数据进行数据挖掘是数据挖掘的研究热点。本章对复杂结构数据挖掘进行了简要介绍,其中关于文本数据挖掘介绍了文本数据的特点、文本挖掘的定义、主要任务、一般过程及其应用;关于 Web 挖掘介绍了 Web 数据的特点、Web 挖掘的定义、分类、一般过程及其应用;关于空间数据挖掘介绍了空间数据的复杂特性、空间数据挖掘的定义、类型和主要用途;关于多媒体挖掘则简要介绍了概念和分类。

习题 11

1. 非结构数据挖掘和结构数据挖掘有何异同?
2. 请比较 Web 挖掘的 3 种方法的特点。
3. 请说明 Web 内容挖掘和 Web 结构挖掘的任务。
4. 多媒体数据挖掘的概念、分类和体系结构。
5. 请简述一下空间数据挖掘的概念、分类和体系结构。

参 考 文 献

[1] 安淑芝. 数据仓库和数据挖掘技术[M]. 北京：清华大学出版社，2006.
[2] INMON W H. 数据仓库[M]. 北京：机械工业出版社，2002.
[3] 王珊，等. 数据仓库技术与联机分析处理[M]. 北京：科学出版社，1998.
[4] 陈文伟，黄金才. 数据仓库与数据挖掘[M]. 北京：人民邮电出版社，2004.
[5] 史忠植. 知识发现[M]. 北京：清华大学出版社，2002.
[6] 徐洁馨. 数据仓库与决策支持系统[M]. 北京：科学出版社，2005.
[7] 王国胤. Rough 集推理论与知识获取[M]. 西安：西安交通大学出版社，2001.
[8] Han Jiawei，KAMBER M. 数据挖掘概念与技术[M]. 北京：机械工业出版社，2006.
[9] MITCHELL. 机器学习[M]. 北京：机械工业出版社，2003.
[10] 王欣，徐腾飞，唐连章. SQL Server 2005 数据挖掘实例分析[M]. 北京：中国水利水电出版社，2003.
[11] 夏火松，等. 数据仓库与数据挖掘技术[M]. 北京：科学出版社，2009.
[12] 何玉洁，张俊超. 数据仓库与 OLAP 实践教程[M]. 北京：清华大学出版社，2008.
[13] 张兴会，等. 基于递阶对角神经网络的失业预测研究[J]. 数量经济技术经济研究，2002，19(9)：114-117.
[14] 张兴会，等. 主成分分析法在神经网络经济预测中的应用[J]. 数量经济技术经济研究，2002，19(4)：22-125.
[15] 张兴会，等. 基于对角 Elman 神经网络的失业智能预测模型[J]. 南开大学学报，2002，35(2)：60-64.
[16] 张兴会，等. 基于神经网络模型的非线性多步预测学习控制器[J]. 控制与决策，2002，17：820-823.
[17] 张兴会，等. 基于神经网络误差补偿的混沌系统广义预测控制[J]. 南开大学学报，2004，37(2)：89-92.
[18] 刘玲，张兴会，梁伟. 一种基于 Aihara 神经元的改进 CBAM 模型[J]. 天津工程师范学院学报，2009，19(2)：12-14.
[19] Du Shengzhi，Chen Zengqiang，Yuan Zhuzhi，et al. Sensitivity to noise in bi-directional associative memory(BAM) [J]. IEEE Trans. on NEURAL NETWORKS，2005，16(7)：887-898.
[20] 刘玲，张兴会. 基于神经网络的数据挖掘算法研究[J]. 计算机工程与应用，2008(9)，563-564.
[21] 王明春，等. 基于相对距离的改进粗 k-means 方法[J]. 计算机应用，2009(4)：1102-1105.
[22] 王明春，王正欧. 一种基于 CHI 值特征选取的粗糙集文本分类规则抽取方法[J]. 计算机应用，2005(5)：1026-1028.
[23] 王明春，王正欧. 基于粗糙集和遗传算法相结合的文本模糊聚类方法[J]. 电子信息学报，2005(4)：548-551.
[24] ZhengXiaoyan，SunJizhou. Finding Frequent Item Sets from Sparse Matrix[C]. 2009 International Conference on Electronic Computer Technology.
[25] 郑晓艳，石连栓，孙济洲. 基于 VOPP 并行编程环境的最大频繁项集生成方法[J]. 计算机应用研究，2009(4).
[26] Tong Yongmu，Zheng Xiaoyan. An Algorithm to Construct FP-Tree on VODCA[C]. 2010 Third International Conference on Intelligent Computation Technology and Automation.